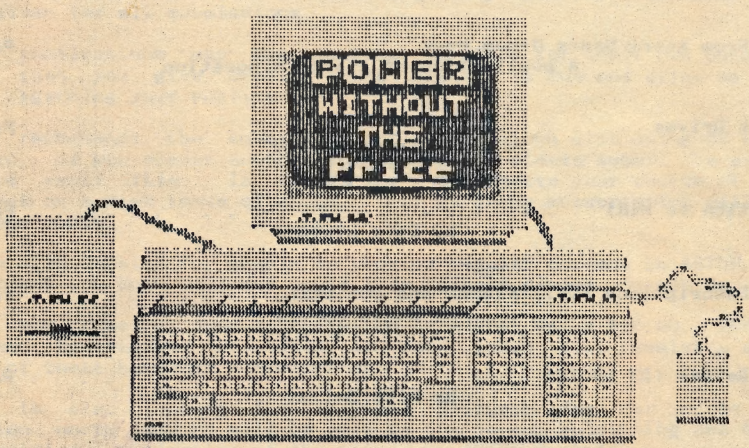
A S T U N

ATARI ST USERS NEWS



EUREKA, CA
AUG 12 2
PM
1986
95501

JULY - AUGUST 1986

$3.00

MAIL EARLY-
ALWAYS US
ZIP CO

ATTENTION USERS GROUP OFFICIALS

This is a complimentary copy of ASTUN. Future copies of our newsletter will be sent free to all groups that are willing to exchange their newsletters with us.

TABLE OF CONTENTS

A C Programing Example (C Tutorial Continues) . . . . . Page 3
by Ralph Porter

Atari ST Tips And Tricks . . . . . Page 19
By Gordon Billingsley, Robert Mann and Brian Duggan

Atari ST Users News Information, Catalog and Order Form . . . . . Page 39
by Ralph Porter

Editors Page . . . . . Page 2
By Ralph Porter

Excerts From Atari Users Group News . . . . . Page 22
A publication of Atari Corporation

Hard Disk Drives . . . . . Page 12
by Eric Lindsey

Introduction to Midi . . . . . Page 15
by Eric Lindsey

Product Descriptions (DFT, KISSED and DOS SHELL) . . . . . Page 27
by Ralph Porter

Program Review (DX DROID) . . . . . Page 30
by Mike Selley

Program Review (THE MUSIC STUDIO) . . . . . Page 28
by Br. Tabias S. Haller, BSG

Public Domain Program Documentation (FDI, A Floppy Disk Index) . . Page 34
by Rod Waehner

Public Domain Program Documentation (TINY STUFF and TINY VEIW) . . Page 37
by David Mumper

ST Update (Excerts From Current Notes) . . . . . Page 17
by Joe Waters and Frank Sommers

The Computer Handyman . . . . . Page 21
by J. Pazina

The Travials of a Novice C Programer (A Modem Utility) . . . . . Page 10
by Brad Davis

ATTENTION USERS GROUP OFFICIALS

This is a complimentary copy of ASTUN. Future copies of our newsletter will be sent free to all groups that are willing to exchange their newsletters with us.

EDITORS PAGE

Hello again. This month I would like to make an announcement concerning the future of ASTUN. I hope you will approve of the changes that will be made. First ASTUN has affiliated itself with STUN (ST Users Network). STUN is presently a few small users groups with plans for future expansion. It plans to form a network of ST users groups throughout the world. The network will exchange public domain programs and club newsletters. ASTUN will bring you the best from these clubs as well as inform you what they are doing.

This magazine will be available through most affiliate groups. Look for a new section coming soon dedicated to Local Atari ST computer clubs.

Presently ASTUN is a small publication with a very small circulation. Sales of the magazine do not cover the expenses of printing and distributing it. Copy fees received for public domain disks have helped to fill the void but just barely. It is hoped that STUN will help to bring a profiet to this publication. We also need more and better quality material. Remember A free issue is given for all submissions.

Other publications may use material in ASTUN for their own purpose. We only ask that you give credit where credit is due and allow us the same privilege regarding your publication.

ASTUN recommends the submission of material on disk using St Writer or First Word. If you prefer another word processer please submit the article as a standard ascii file. In return you will receive your choice of a public domain disk or a disk issue of ASTUN. Your hardcopy subscription will also be extended one month.

We are also looking for people to provide regular columns in ASTUN. If you have a subject you know well and would like to help please contact us.

Would you like a free subscription to ASTUN? Value $15.00. The article judged best by readers of ASTUN in each issue will receive a six issue extension of their hardcopy subscription.

Astun is also looking for original pictures for our cover. A Full subscription worth $40.00 will be paid to the person submitting the best logo for our magizine. (Selection of the winner by ASTUN'S staff).

We want to improve our appearance in the coming months. We hope to enlarge our present magazine and expand to a monthly publication. If you have a talent and can offer anything that will be of use to other readers pleas help us out. Lets make it work so all ASTUN readers and Atari ST users can benefit.

ASTUN is no longer going to offer a disk only subscription. The remaining subscription of the few who asked for it will be honored.

If you can offer any assistance to this magazine or just wish to tell us your feelings on our effort please let us know. We need your input to make this magazine go. We need to know what you want to see within its covers.

You will notice with this issue a change in your mailing label. Next to your name is the expiration date of your current subscription. The letter following the expiration date indicates the type of subscription. A "H" indicates Hardcopy, A "D" a disk subscription. Some other letters may be used for sample copies and internal use. Please let us know if the information is not correct.

A C PROGRAMING EXAMPLE

In prior instalments of this tutorial we discussed various aspects of C programing theory. In this installment we will present a few C functions and explain their purpose along with a detailed description of how they work. Future installments will most likely be less detailed.

The code presented is for the Atari Development System, however few if anychanges should be needed for a different compiler. No floating point orstructures are used in this installment.

The code chosen may not be the most efficient but it represents several important programing concepts. The line numbers in the program listing are for reference only. C does not use line numbers. If you type this code into your computer do not type in the line numbers. Lines 1 and 2 of the source code are preprocessor include statements. They tell the compiler to include the specified files in the compilation process. The first file stdio.h contains macros for standard input and output. Almost all C programs include this file. The second osbind.h includes macros specific to the Atari operating system. This includes bios, extended bios and gem calls.

Line 4 contains a define pre processor statement. In this case it defines a constant which represents the maximum length of a string the program will allow. The rest of the define statements are defining macros. When the compiler sees a defined constant or macro the code or value to the right of the defined key word is substituted for the key word itself. It can be used to simplify constants by giving them a name, and to shorten keystroke as is the case with macros. In the example on line 7 everywhere the programer inputs the statement getchar(), the statement actually compiled is Cconin(). One may wonder the value of this. It may in some cases seem meaningless. In others like in lines 12 and 13 it saves the programmer numerous keystrokes. In many cases a macro is faster than calling a function because the actual code is executed in place. Our example on line 7 is for a different reason. Notice the undefine pre processor statement in line 6. The function getchar() is defined in the stdio.h file. The problem is it does not always work properly due to a bug in the development system compiler. This is also true for gets() which is a true function. The gem call Cconin() works like the getchar() function should. Using it here in this macro allows the programmer to write code that is transportable to compilers that work correctly. In the working compilers the only change needed will be to remove the related define statement from this source file. The gets() on line 8 replaces the call to gets() with a call to Getstr() a function that we will shortly discuss. This replaced function again works like gets() should but does not work in the development system. One other defective function is scanf(). The fix for it is not as easy due to the complexity of the function. To get around it read in a string with the gets() call and then do a sscanf() function on the resulting string.

The problems discussed in the development system standard i/o library are caused because the C language expects a new line character (control j) rather than a return to end the line. The C run time library should have fixed this problem but it didn't. The standard functions due to this error do not return control to the calling program when return is hit. To return control on the ST you must hold control down while pressing the j key. The macro replacing getchar() and the function call replacing gets() fixes these problems so the standard return key can be used.

Lines 9 through 13 are true macros. These macros are normally contained in a file ctype.h. This file was missing in early versions of the development system. We elected to define them here to show how macro can be used. The actual macros in the ctype.h file are different. This is because a structure is used to define character types. The final result however is the same. In our case they are stand alone macros.

The purpose of the first three are similar. They perform a test to see if the character under test is a digit (number) or an upper or lower case character respectively. These macros return a true or false condition when executed. The macros on lines 12 and 13 have an internal call to another macro. They return the character either as tested or else converted to the opposite case, depending on the result of the internal macro call. This macro uses the C conditional operator "?:". This operator reads like this. If the condition on the left of the "?" is true then perform the operation to the

right of the "?" and the left of the ":". If the test condition is false perform the operation to the right of the ":". It is a form of if then statement and is very useful if only two choices are present.

The function defined in lines 15 to 22 fixes the problem with the gets() function described earlier. As in all C functions a function name is given along with the variables it expects. Here the function name is getstr and it expects one variable. The second line tells us it is expecting a pointer to a character or character string. The "*" means a pointer to int. The definition phase as used here. It could also mean the contents of if used outside of the definition. Line 17 the "{" delimits the start of the function. The "}" at line 23 is the end of it. Line 18 defines a character array of 82 characters. An array of characters is refered to as a string. The constant MAXLEN defined back on line 4 is 80. Line 19 places the number 80 into the first element of the string. This is required by the gem call we are about to make. The second element of the string will contain the actual number of characters returned when the call is returned. For our purpose here we will ignore this element. The third through n'th characters is the actual string returned. The call we use to accomplish this is Cconrs(). This function works similar to the way gets()should except for the extra two characters needed to begin the string. The last line of the function copies the returned string into the variable pointed to bys. Upon return to the calling routine the keyboard input is effectively returned.

Back to the define statements. They have all been explained now with the exception on line 5. getkey() will work similar to getchar(). The difference is that no echoing of the character takes place and that no character translation takes place. This function will let you look for keys that are normally intercepted by the operating system such as (control c, s and q). We have included this macro for your use. It is not used in our examples today.

Lines 24 through 113 contain a series of functions that make up a simple utility designed to make data entry more user friendly.

When entering a date most computer programs require that you follow an exact method when inputting. This utility removes the need for specific formats. Regardless of how you input the date the result is the same. The returned date is the format that the programmer needs. The only rule is that the date must be input in a non- military fashion. This means month first followed by day and year. These are all legal input. (January 11,1951 Jan 11, 1951 1/11/51 1-11-51 01 11 1951). Many other combinations and punctuation may beused if desired by the inputter. All examples used return the date in the format (01-11-51).

In addition to making it easier to input dates this function does a check to insure that the entered date is a valid date. If an invalid date is entered it reprompts you until you enter a valid one. The program properly handles leapyear.

Whether you have use for this utility or not is not important. Theimportant thing is that several programing techniques will be illustrated.

Before we get to the specifics of this utility we must discuss how to call it. In an earlier tutorial we noted that all C programs must contain a function main(). In our sample this is at line 116. Note that the placement of this function is not important. It may be at the beginning, the end, or anywhere in between. What is important is that it is always where the program will begin executing.

We will now walk through the function main(). Before we do however we must point out that the statement on line 24 declares a global variable. The variable is a pointer to an array of strings that contain the months of the year. It is initialized in this statement. Because this is defined outside of any function this is a global variable and is available any place in the program. Had we defined it as a static variable at this point it would only be available to this source file. Note that two lines are used in initilizing the variable. C is a punctuation delimited language. Code can easily be spread over several lines. The symbals "{" and "}" are used for delimiting.

Getting back to the program the first thing we do is define variables c, send date as characters. c is a single character, s and date are pointers to a character string or a character array. All arrays define pointers even though the "*" pointer symbal is not used. We then define integer variables i, month, day and year. These are all automatic variables and none of them are

initialized. This means they may contain unwanted data. It also means that they will go away when the function they are defined in finished executing. Had we declared them as static then they would automatically be initialized to zero and would retain their value whenever the function containing them was called. Static variables have no advantage here so we did not use them. We only used one global variable so we could illustrate some variable passing to functions. One should be careful in the use of global variables as they can be a source of hard to find program bugs.

Line 120 and 121 call the most common C function. Here we simply print the quoted strings to the output device or C.R.T. The "!n" is the new line character and must be included if we desire one. Line 122 calls getchar(). In our program it actually calls Cconin() because of the earlier defined macro on line 7. This function returns a single key as you press it. It echoes the key pressed to the terminal. Line 123 shows the conversion ability of printf. The% indicates a variable, in this case d, representing a decimal integer. The item to be output is the variable c that was previously input. The decimal value of a character is its ascii value. Lines 124 to 127 demonstrate the working of the gets() macro. It asks you to input a string ended with a carriage return. It then uses the c function strlen() to tell you how many keys you pressed. Notice that this function is nested inside of the printf function. This capability is one reason C is such a powerful language. Not many languages allow this form of nesting.

The next line calls the first of the functions making up the getdate utility. It shifts execution to line # 40. It passes to this function the address of the date variable. (Pointers store addresses). The function is expecting a character pointer and that is what we sent it. This function prompts the user for the date and places the users input into the date variable. It then calls the function checkdate() which will reformat it and verify its a valid date. Again the date pointer is passed. Within the function three automatic variables are defined. These variables (month,day,and year) are not the same as the variables with the same name in main. Remember autom aticvariables are local to the function they are defined in. This function first calls the macro isdigit() passing it the first character of the date string. If the first character is a number between 0 and 9 the sscanf function is called placing the value of the first number into month variable. sscanf() like printf can convert data types. In this case a string is converted to a int.

If the isdigit test failed the else statement is executed. Note only one line of code or instruction followed the if. No delimiting is needed if only one instruction is to be executed in comparison and looping tests. When more than one item is to be performed then delimiting must be done as with this else statement. All commands within the braces will be executed.

The next three instructions (lines 46-48) force case conversion of the input string to match that of the test strings define in line 24.

A loop is then executed up to 10 times looking for a match. The strcmp() function checks two strings to see if they are equal. We use the strncmp() function that compares two strings to the specified length for equality. Here we test only the first three characters. If the first three characters match the break statement is executed which dumps us out of the loop. If they don't the loop is executed again. The for statement limits the loop to 12 passes. C does not use a next statement. All parameters for a for loop is contained in the for statement. A for statement has three parts. The first allows setting variables to assigned values. More than one variable may be assigned by delimiting them with comas. The parts of the for statement must be delimited with semicoluns. The middle of the loop contains the exit conditioner test statement. In this case the test is made to see how many passes took place. The last item is what to do at the end of each loop before the next one starts. Here we increment the test variable. The ++ in C is the same as saying X=X+1 in basic. An interesting sideline is this. If the ++ appears on the right of the variable then the incrementation takes place after it is used. if it precedes the variable the incrementation takes place before it is used.

All arrays in C begin with element 0. That is why we now increment the month by one so it will match the calendar month. At this point if a valid month was entered it will be in month. If not month will contain 13.

Line 55 calls the function nxt_fld() and passes it the date pointer. Note the underscore in the function name. This is used to make variables more

descriptive. Use it where a space would normally be used.

The purpose of nxt_fld() is to remove any data that has already been obtained and to remove punctuation from the test string. Note we passed the date pointer but the function expects a pointer to s. The data passed to C functionsis always a copy of the original. The original is not changed. Passing pointers or addresses allows the program to change the contents of those pointers effectively changing the data. If no pointer is passed the original data cannot be changed within the function. This s within the function now contains the save value that is in the date variable outside of the function. This feature of C eliminates many errors that creep into languages like basic that define their variables globally.

We first declare the integer variable i and assign it a value of 2. Then we execute the while statement. While is another form of looping. The statement executes until the test condition is false. The "!" or not operator reverses the test results. It reads while the i'th element of string s is not a digit execute the loop. When a digit is reached the loop exits. We then copy the string with the strcpy function. The first part of the string is lost.
Since s is a pointer the contents of the date string that was mapped into s is changed. The reason this function is a function is that the program uses it twice. By making it a function the code need not be duplicated.

If you are confused about variable passing you may need to review past tutorials or consult a book on C. A function can return one value. If more than one is needed then it must be done by passing pointers.

After finishing the nxt_fld() function we return to the line following its call. We now read the day with sscanf, again call nxt=fld and then read the year. Note the "$" in the sscanf calls. The "$" means the address of. sscanf() can assign many variables so the addresses or pointer must be passed. sscanf() returns the number of variables read. We are ignoring this in our routine.

The "=%" in the next line may need explaining. This line is saying (year= year % 100) where % means modula or remainder. This forces a four digit year into a two digit one.

The next several lines do error checking. We use a switch() statement. The switch is a multiple if else derivative. We first check for a valid month and then a valid day. If the month is February we need to make more checks. If it's February and the day is 29 we need to know if it's leap year. If its not February we need to know if its a month with only 30 days. All checks for this take place here. If any test fails the error routine is executed. The function date_err() prints the reason for the error and again calls the getdate() function. The process continues until a valid date is input.

Note the recursion. getdate() calls date_err() which in tern calls getdate(). Recursion can be used to advantage in C. In some cases a function may call itself. In a future tutorial we will probably demonstrate its use.

With a valid date entered the sprintf() command reconstructs the date to the form desired by this programer.

With the date input, reformated and validated we now return to main. We print the reformated date. The next line shows how we can use scanf to extract information from the standard form for use in the printf() statement on the next line which prints the date in a longhand form. Note the "%*" this tells sscanf to skip over the data specified rather than assigning it.

We leave the rest of the program up to you. You should be able to follow it now.

Now some suggestions for your own experimenting. The function could be modified to get the date from the ST keyboard clock. A bios call exists for this. You may wish to alter the date format I chose.

I Intentionally left two potential problems in this code. Both involve the user inputing to little information when the date is requested. One is leaving out a field such as the year. The other involving the year. The coding of the program will only be valid if a two or four digit year is entered. What changes are needed to fix this.

Finally a request. If you can use what you know or what you are learning to develop functions that would be useful to other ST users please send them to us. If we publish the function you will be given full credit. The issue containing the function will be yours free of charge. Your subscription will be extend one month to do this.

```
000
001 #include <stdio.h>
002 #include <osbind.h>
003
004 #define MAXLEN 80
005 #define getkey() Cnecin()
006 #undef  getchar()
007 #define getchar() Cconin()
008 #define gets(s) Getstr(s)
009 #define isdigit(c) ((c) >= '0' && (c) <= '9')
010 #define isupper(c) ((c) >= 'A' && (c) <= 'Z')
011 #define islower(c) ((c) >= 'a' && (c) <= 'z')
012 #define tolower(c) (isupper(c) ? (c)+('a'-'A') : (c))
013 #define toupper(c) (islower(c) ? (c)+('A'-'a') : (c))
014
015 Getstr(s)
016 char *s;
017     {
018      char a[MAXLEN + 2];
019      a[0] = MAXLEN;
020      Cconrs(a);
021      strcpy(s,a + 2);
022 }
023
024 char montha[12][10] = {"January","Febuary","March","April","May","June",
025           "July",  "August","September","October","November","December"};
027
028 #define MONTH (month -1)
029 #define YEAR  (year + 1900)
030
031 Getdate(date)
032 char *date;
033 {
034      printf("Enter Date: ");
035      gets(date);
036      Checkdate(date);
037      putchar('\n');
038 }
039
040 Checkdate(date)
041 char *date;
042 {
043      int month, day, year;
044      if(isdigit(date[0]))
045           sscanf(date,"%d",&month);
046      else
047      {
048           date[0] = toupper(date[0]);
049           date[1] = tolower(date[1]);
050           date[2] = tolower(date[2]);
051           for(month=0;month<12;month++)
052                if(!strncmp(date,montha[month],3))
053                     break;
054           month++;
055      }
056      nxt_fld(date);
057      sscanf(date,"%d",&day);
058      nxt_fld(date);
059      sscanf(date,"%d",&year);
060      year %= 100;
070      if(month < 1 || month > 12)
071           date_err(0,date);
072      else if(day < 1 || day > 31)
```

```
073             date_err(1,date);
074         else if(day > 28)
075             switch(month)
076             {
077             case 2:
078                 if(day > 29)
079                     date_err(2,date);
080                 else if(day == 29)
081                     if(((YEAR % 4) > 0)
082                      || (((YEAR % 100) == 0) && ((YEAR % 400) > 0)))
083                         date_err(3,date);
084                 break;
085             case 4: case 6: case 9: case 11:
086                 if(day > 30)
087                     date_err(2,date);
088             }
089         sprintf(date,"%02d-%02d-%02d",month,day,year);
090 }
091
092 nxt_fld(s)
093 char *s;
094 {
095         int i = 2;
096         while(!isdigit(s[i]))
097             i++;
098         strcpy(s,s + i);
099 }
100
101 date_err(err,date)
102 int err;
103 char *date;
104 {
105         switch(err)
106         {
107         case 0:
108             printf("\nInvalid Month.\n");
109             break;
110         case 1:
111             printf("\nInvalid Day.\n");
112             break;
113         case 2:
114             printf("\nTo Many days for entered month.\n");
115             break;
116         case 3:
117             printf("\nEntered year is not a leap year.\n");
118             break;
119         default:
120             printf("\nInvalid date.\n");
121         }
122         Getdate(date);
123 }
124
125
126 main()
127 {
128         char c, s[80], date[30];
129         int i, month, day, year;
130         printf("\n\n\n\nThis is a test file\n");
131         printf("Type a single character: ");
132         c = getchar();
133         printf(" has an ascii value of %d.\n",c);
134         printf("Enter a line of text: ");
135         gets(s);
136         printf("\nYou Typed %d characters.\n",strlen(s));
137         Getdate(date);
```

```
138       printf("The standard form for the date entered is %s.\n",date);
139       sscanf(date,"%d%*c%d%*c%d",&month,&day,&year);
140       printf("Expanded it is %s %d, %d.\n",montha[MONTH],day,YEAR);
141       printf("\nTHANK YOU\n");
142        printf("\n\nHit return to end: ");
143       c = getchar();
144 }
```

ST USER'S NETWORK

The ST User's Network is a new organization founded to expand awareness concerning the Atari ST computers. We are dedicated to bringing quality services, authorative instruction and friendly socialization to the greater Wasatch Front. Soon to expand around the country and the world.

We currently feature:

Monthly meetings on the third Wednesday of each month.

Two nationally affiliated electronic B.B.S.'s.
Featuring 60 Mega Bytes of online storage each.

The Atari St User's News (A.S.T.U.N.) is optional.
A magazine dedicated to the Atari ST computer.

Monthly public domain disk issue available each month for only $4.00, packed with useful prg.'s. (Price is ~~$5.00~~ by mail).

Public Domain software library available online and through a club librarian.

The S.T.U.N. is ST specific. We will support subsequent ST type computers, however we will not support older Atari computers. We offer a large and rapidly expanding techinal and general library, intensive SIG type classes offered for a wide variety of subjects and dynamic meetings featuring hands on demonstrations and techinal discussions. We welcome all ST user's. Please call or write for more information.

P.O. Box 27285
Salt Lake City, Utah 84127
801-487-8648
801-487-1762
The H.O.G. 964-9764 (BBS) 24 Hours a day
The W.O.S. 485-6349 (BBS) 24 Hours a day

THE TRAVAILS OF A NOVICE 'C' PROGRAMMER

by Brad Davis

I've had my ST for about 6 months now, and I think I have the basics pretty well in hand. I began learning 'C' two or three months ago at the urging of all my more adept user/friends, and so far I am impressed with the power and versatility of the language. It would seem natural that when I encountered my first real hardware compatibility problem that I would use 'C' to come up with a suitable software solution, right? Right! So I got out my flowcharting template (good programmers always flowchart, so I learned way back in FORTRAN IV class) and sat down to work out the program logic. Then with the flowchart done, I sat at the console and merrily began to enter code. 637 compilations later, I actually had a working program. Never mind that it didn't work, it actually compiled and linked! With that large accomplishment under my belt, I bent to the task of making the darn thing do what I had intended it to do. Suffice it to say that it took two months and much head scratching, but I finally came up with a program that worked and was aesthetically pleasing.

The program I wrote is called DON'T ANSWER!. This little (less than 6,000 bytes) program disables auto answer on any Hayes compatible modem using the S0 parameter. OK, I can see some of you out there saying "why do all that when you can just flip a switch?!". Remember I said I had a hardware compatibility problem? The problem is that I have a Habamodem. Habamodems do not have switches. That's one reason it's less expensive than some other Hayes compatibles. It's a good modem, I just got tired of explaining why people heard an obnoxious whistling when they called my house. You see, I have all my computer equipment tied into a master switch so it all comes on simultaneously. And for those of you saying "why not just switch the modem off when not in use?", the Habamodem also lacks an on/off switch. It's either plugged in and running or unplugged and off. With these limitations the simplest approach to accomplish what I wanted seemed to be an autobooting utility that sent a command to the modem to turn auto-answer off. Then I could receive phone calls without having the modem beat me to the phone, and use the modem without a lot of plugging and unplugging.

DON'T ANSWER! has simple program logic (see flowchart). TOS is used because the program is meant to be used in an AUTO folder. Auto booting programs are run before GEM is loaded, and so wouldn't take advantage of GEM features anyway. Credits are displayed first, the program clears the RS232 buffer to avoid a backup in the buffer, the command ATS0=0 is sent to the modem to disable auto-answer, then the program displays a message to tell the user that the feature is disabled and ends. A look at the source code shows the program execution, and the comments explain what is happening.

This program is my first attempt at manipulating the RS232 port of the ST. At first, I thought that there was too much left to the programmer, as I was used to an 8 bit Z80 machine that pretty much did everything 'automagically'. However, as I began to understand the workings of the port, I realized that the designers were trading a little ease of use in the beginning for a lot of power and flexibility in later usage. This utility can easily be expanded into a mini-BBS that could be run from the desktop (a project I hope to attempt in the near future), sort of like a computer answering machine to take messages and automatically transfer files. The basics of telecommunications are all here. The Bconstat, Bconin and Bconout functions make up the major portion of any telecommunications program. These commands, a few auxiliary ones to control peripheral I/O and some study should allow anyone to write a telecommunications application.

This application should be a welcome sight for those Habamodem users who use a single phone line for both voice and modem calls. It solved my problem quite nicely, and was an excellent programming lesson.

```
/*Don't Answer! Utility*/
/*by Brad Davis PA-C CIS#72335,267*/
/*This little utility in an auto folder will set Hayes compatible*/
/*modems w/o switches to _NOT_ answer the phone*/

#include <stdio.h>  /*I/O conventions*/
#include <osbind.h> /*Binder to talk to OS*/
#define RS232 1     /*Device spec for RS232 I/O*/
#define baud 7      /*RS232 Configurations...*/
#define ctrl 1
#define ucr -1
#define rsr -1
#define tsr -1
#define scr -1
int wait;           /*variable for RS232 input*/
int n;              /*Count variable*/
char ltr_in;        /*variable for RS232 input*/
char *ltr_out=" ATS0=0\r"; /*String for RS232 output (turns A/A OFF)*/

main()
{
    printf("DON'T ANSWER!\n");
    printf("by Brad Davis PA-C\n");
    printf("(c)1986 DON'T PANIC! Inc.\n");
    Rsconf(baud, ctrl, ucr, rsr, tsr, scr);
    clr_buffr();
    printf("\n");
    put_it();
    printf("\n \n");
    printf("\nAuto Answer _disabled_\n");
    printf("Bye....."); /*The END!*/
}

/*Now for the routines that do the dirty work...*/

clr_buffr()
 {
  while((wait=Bconstat(RS232))!=0)
   {
    ltr_in=Bconin(RS232);  /*read the buffer just to clear it*/
   }
 }

put_it()
 {
      while(*(ltr_out)!='\0')  /*Loop to get the output string*/
       {
          Bconout(RS232, *(ltr_out));      /*Send char out to the RS232*/
          ltr_in=Bconin(RS232);            /*then look for the echo from
                                             modem*/
          (ltr_out++);      /*then increment until the sucker is done!*/
       }
 }  /*and...back to caller....*/
/*That's ALL, folks!*/
```

HARD DISK DRIVES
by Eric Lindsay
A Reprint taken from the Jackintosh Gazette

HISTORICAL INTRODUCTION

The hard disk drives used by mainframes generally have multiple disk surfaces, and multiple read write heads working simultaneously. This allows storage of all the data lines from a mainframe memory at one time. Most such drives also have removable disk packs, and elaborate air filters.

Microcomputers typically use a much simpler (and cheaper) system, almost all of which are "winchester" style fixed drives, from which the disk pac can not be removed. The Winchester name comes from the code name of an early project to develop cheaper drives. The differences are that, being a fixed disk, engineering tolerances are slightly eased. Also, although there is still a separate read write head for each disk surface, only one head is used at a time, and thus only one read write amplifier and associated electronics is required. Also, it means as few as two or four disk surfaces can be included, rather than the 16 or 32 of a mainframe hard disk. Two disadvantages of this are that data must be converted from parallel into serial form before reaching the head for writing, and the reverse when reading. This also means that a Winchester drive must run a lot slower than a mainframe hard disk, since data reaches it in a serial stream. The slowness is relative; it is still many times faster than a floppy disk drive, since most hard disks rotate at 3600 rpm, rather than the 300 rpm of a typical floppy drive.

There are a large number of "standards" in the Winchester world, however we will consider only the most common of these. A complete Winchester system consists of three conceptual blocks which are mostly separate, although they can be built on a single board nowadays, and will increasingly be built in one piece. These are the Host Computer Adapter, the SASI or SCSI Drive Controller and Interface, and the ST506 of similar disk drive interface together with the actual drive. The physical size, capacity, and to some extent, speed of the drive doesn't really make much difference to the way things work.

The Host Computer Adapter (HCA) provides electrical and signal interfaces between your computer and the SASI bus. In the Atari 520ST, it consists of a Direct Memory Access (DMA) port, plus relatively few control lines, and is built into the ST. A similar device is now built into the new Mackintosh Plus. In the IBM PC XT the HCA and the hard disk controller are built into their Fixed Disk Adapter card. Apart from transferring data under DMA, the HCA is only required to pass a software command descriptor block (CDB) to the drive controller, using a particular software protocol. Although there is provision for 8 classes of CDB, 0 through 7, only three affect normal disk operation. Class 0 is data, non-data and status commands, and consists of siz bites. Class 1 is disk copy commands, and class 6 is disk parameter assign commands. Class 7 is for diagnostics, and the other classes are reserved for future use. The command protocol is sufficiently flexible to allow for the use of optical disks, and other intelligent peripherals in the future. For example, we are considering the possibility of connection the Ozi Rabble computer as an intelligent household control computer.

Since the Drive Controller must respond to software commands, it follows that it must be "intelligent", or consist of a complete microcomputer itself. It also needs some sort of relatively standard hardware and software interface to the Host Computer Adapter. One of the earliest intelligent standards for the micro world was the Shugart Associates System Interface (SASI), devised by the Xerox owned, but now defunct, drive maker Shugart Associates around 1980. The question of which "standard" to use led to a division among members of the Americal National Standards Institute committee (ANSI X3T9.2). The mainframe and mini computer members of which tended to favor the Intelligent Peripheral Interface (IPI) based on a Sperry Univac design. The breakaway commitee (ASNI X3T9.2) favored the Small Computer Systems Interface (SCSI), which is essentially the same as SASI.

Hard disk drives have been decreasing in physical size, and the interfaces to the disk drive controller have also changed. The older 14 inch drives tend to use Control Data's Storage Module Drive (SMD) and MMD interfaces, while the 8 inch drives mostly use the SA1000 interface. With the arrival of five and a quarter inch and smaller hard disks, most makers followed the Seagate ST506 or ST412 interface to the drive controller. Interestingly, the ST506 is a spinoff

of the Shugart Associates SA1000 eight inch interface, except the data rate is increased form 4.34 Mbits/s to 5 Mbits/s. Many (but not all) ST506 disk drive controllers are able to handle the SA1000 style drives also.

One problem with the ST506 interface is that it calls for the drive data to be supplied in Modified Frequency Modulsted (MFM) form, somewhat similar to that provided by a double density floppy drive. This was originally used because it saved space on the drive electronics card, where space was at a premium. The data seporator, which consists of a Phased Locked Loop (PLL) with a large number of components, including fine tuning potentiometers, is left on the disk drive controller card, where there is room. However there are disadvantages to this approach, as the system is susceptible to noise induced errors. Mounting a data separator in the drive itself eliminates this problem, as the drive then produces separated data and a synchronising clock signal for the controller. Several drive manufacturers have proposed a modified ST506 specification to allow this. This is the Enhanced Small Disk Interface (ESDI). The other advantages of this ESDI approach is that it allows faster data rates up to 10Mbits/s, since PLL can be tuned to suit the specific drive rather than always set for 5 Mbits/s. This also makes it possible to use run length limited (RLL) encoding schemes such as IBM's 2.7 code, or Apples equivalent. These can allow much greater data densities, and thus higher capacities while still using the same style of drive.

ST506 INTERFACE AND MINI WINI DRIVES

Both the ST506 and ESDI interface include a 34 pin connector, in which the odd numbered pins are ground returns. The even pins on J1 are as follows:

PIN	ST506	EDSI
02	Reduced Write Current	Head Select 3
04	Reserved (Head 2)	Head Select 3
06	Write Gate	Write Gate
08	Seek Completed	Seek Completed/Config Status Data
10	Track 0	Track 0.Transfer Acknowledge
12	Write Fault	Write Fault
14	Head Select 0	Head Select 0
16	Reserved (J2 pin 7)	Sector
18	Head Select 1	Head Select 1
20	Index	Index
22	Ready	Ready
24	Step	Step/Transfer Request
26	Drive Select 1	Drive Select 1
28	Drive Select 2	Drive Select 2
30	Drive Select 3	Drive Select 3
32	Drive Select 4	Read gate
34	Direction In	Direction In/Command Data

The twenty pin connecter J2 is very different in both.

PIN	ST506	EDSI
01	Drive Selected	Drive Selected
02	Ground Return	Byte Clock
03	Reserved	Seek Completed
04	Ground Return	Address Mask Enable
05	Reserved	Write Protected
07	Reserved (to j1 Pin 16)	+- Write Clock
08	Ground Return	+- Write Clock
09	Reserved	Cartridge Changed
10	Reserved	+- Read/Reference Clock
11	Ground Return	+- Read/Reference Clock
13	+ MFM Write Data	+- NRZ Write Data
14	- MFM Write Data	+- NRZ Write Data
15	Ground	Ground
17	+ MFM Read Data	+- NRZ Read Data
18	- MFM Read Data	+- NRZ Read Data
20	Ground Return	Address mark Found

The DC power connector supplies +12 volts on pin 1, with the ground return on pin 2. Pin 4 is the +5 volt supply, with pin 3 the ground return.

The ST506 interface pin outs listed above are fairly standard on many hard disk drives including those supplied for use with the IBM PCXT fixed disk adapter card.

SASI BUS INTERFACE

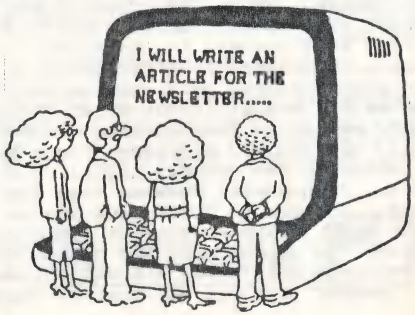
The SASI bus used a 50 pin connector from the Host Computer Adapter to the disk controller. The pin out is as follows. All odd numbered pins are connected to ground.

PIN	Signal	Function	Atari Line	
02	Data 0	Bidirectional Data	1	
04	Data 1	"	2	
06	Data 2	"	3	
08	Data 3	"	4	
10	Data 4	"	5	
12	Data 5	"	6	
14	Data 6	"	7	
16	Data 7	"	8	
18	Data P	Parity Bit, if used		
32	Atn	Attention		
34	Spare	Reserved		
36	Bsy	Busy		
38	Ack	Acknowledge	14	*Ack (Out)
40	Rst	Reset	12	*Reset (Out)
42	Msg	Message		
44	Sel	Select		
46	C/D	Control/Data		
48	Req	Request		
50	I/O	Input/Output	18	

In the Atasi ST pins 11, 13, 15, and 17 are ground.

ATARI HARD DISK PORT

Although I have listed the Atari ST hard disk port lines above as though they are exact equivalents of the SCSI interface, this is not really teh case. Atari has apparently built up their own controller, based on some Western Digital chip set, and have not followed the SCSI interface. Luckily, it does appear possible to interface the SCSI interface and the Atari hard disk port. In a future article we will discuss more on the differences between these ports, and give a detailed description of the SCSI software protocols used to drive a hard disk controller.



Reprinted from December J.A.C.G.

Introduction To Midi

by Eric Lindsay

A reprint taken from the Jackintosh Gazette

produced by Blue Mountains Homebrew Computer Club in Australia

MIDI is an acronym for Musical Instrument Digital Interface, and as that suggests, it is a way of controlling electronic musical instruments by means of a digital signal. The hardware for doing this is very simple, and there is aso a moderately well defined software protocal that has been accepted and used by a number of musical instrument manufacturers. Unfortunately, present prices for MIDI instruments are still rather too high for anyone except dedicated users, however prices are falling, and the interface itself could also be used for computer to computer communications.

MIDI connectors on equipment are 5 pin 180 degree female DIN, with pin 1 being the THRU transmit data, pin 2 the shield ground, pin 3 the THRU loop return, pin 4 the serial data line, and pin 5 the serial loop return line. There are either two or three such connectors on each piece of equipment. The MIDI OUT connector has pin 4 as a transmit line. The THRU connector simply passes along the signal, and is often ommitted from equipment.

Electronically, the MIDI interface appears to be a simple serial current loop, operating at 31,250 Khz!, This can easily be generated by many computers by feeding a 500 khz clock into a 6850 ACIA, or an equivalent chip, and dividing by 16. The output should be via some sort of buffer chip such as a 7406 open collector driver. Input should be via an optoisolator.

Programing the MIDI is something about which i've been able to find no direct information, nor were Yamaha or any other synthesiser manufacturers able to tell us the protocols when we contacted them. However I did locate a simple program for driving a MIDI from a Commodore 64, and based upon the assumption that the serial port is internally similar to a 6850, here is what you do if using a computer with a built in 6850 as the Atari 520ST.

The first step is to initialise the 6850 ACIA, by forcing a reset (binary 0000 0011 into its control register), and then setting it up divide by 16 (binary 01 into control register bits 1 and 0), * bit data with one stop bit, no parity (binary 010 into control registers 7,6,5). It does not appear that the MIDI interface makes any use of any control lines provided by the ACIA, so receiving software MUST be fast enough to intercept all characters transmitted at the full 31,250 KHz. This doesn't matter when we are transmitting, however if writing a "receive" program, it would have to be interrupt driven, and not have too much overhead. You certainly couldn't do it quick enough from basic.

```
10 Def Seg = 16776196 : Rem Set up offset for poke
20 Let Midi-Acia-Ctrl = 0 : Rem ACIA is at $FFF4
30 Let Midi-Acia-Data = 2
40 Poke Midi-Acia-Ctrl, 3 : Rem Reset
50 POKE Midi-Acia-Ctrl, 85 : Rem 8 bit, 1 stop bit
```

Note that what we are trying to do here is Poke 16776176,3:Poke 16776196,85. However there are two potential problems we can't resolve until we actually have a 520ST to play with. It would be preferable to use hexadecimal numbers in the first place, however it appears that these can only be used in the 16 bit range, which is insufficient for our needs. This probably means our variables will be evaluated as real numbers, but we have no idea of the precision of real numbers in the 520ST - they had better be at least 24 bit, but since real numbers in array occupy 4 bytes, they probably are. Also, to peek or poke a single byte, rather than two or four bytes we apparently have to use the Def Seg statement as an offset, but unfortunately this section of the ST Basic manual is inadequately documented.

MIDI notes are actually two events, a trigger for a particular note, and a corresponding release. Without the release signal, the note continues. The MIDI bus can provide up to 16 channels, However some instruments respond only to certain preset channel numbers. This means you can control a number of different instruments at once. The number of notes you can trigger from a particular instrument seems to depend upon the number of voices available to the instrument; if you send too many notes without releasing them, the excess notes will be ignored. I gather that you can control only a single voice with each channel.

The trigger and release each consist of three bytes, which contain a header, a channel number, a note value, and a velocity value.

The first byte is considered as two nibbles. The high bits (4 to 7) are the header. The binary pattern 1001 (decimal 144) triggers a note, while binary 1000 (decimal 128) releases the note. The low bits (0 to 3) provide the channel number, form 0 to 15, however it appears that most MIDI information lists the channel numbers as 1 to 16, so you may have to subtract 1 from the channel number to get the correct figure for coding purposes.

The second byte determines the frequency of the note, and can range from 1 to 127. The frequency increases by a semitone for each digit, so notes 12 apart span the octave, and (if the synthesiser can handle it) you can use 10 octaves. Middle C is produced by sending 60, and has a frequency of 261.6Hz.

The third byte is a "velocity" byte, in the range of 1 to 127, and is ignored by MIDI instruments that do not have a "velocity" or touch sensitive control from their keyboard. Basically, a high number will produce a "harder hit" note. You don't actually need this byte for note release, but a 0 byte is included for symmetry.

Although writing a MIDI demonstration using this should be easy (just send out a whole heap of three byte values, which you could store in data statements), the value of this sort of thing seems rather minor. I gather most synthesisers have additional command available for controlling envelope shapes and other more advanced synthesiser features, however I can't find any information on these. If anyone who owns a MIDI cares to contribute more information, our pages are open. I have no access to a Midi, so I can't check any code even if I wrote it.

Regent Base's procedural language make it a natural for handling any of your small business needs. Modules are available for Invoicing, Accounts Receivable, Checkbook Balancing, General Ledger, etc. Regent Base is a relational database written specifically for the Atari ST. Don't settle for simple clones of IBM products. Regent Base is easy to use and state-of-the-art!

- Command Editor
- Procedural Language
- Templates Available
- Data Entry Editor
- Define Custom Forms
- Custom Reports
- Mutiple Sorting Functions
- B+ Tree Indexing
- Windowing + Mouse Controls
- Hard Disk or Floppy
- Not Copy Protected

Regent Software
7131 Owensmouth, Suite 45A
Canoga Park, CA 91303

Phone: (818) 883-0951
Telex: 5106011374 REGENT UQ
CompuServe: 72457,3171

==========
ST UPDATE:
==========

by Joe Waters and Frank Sommers
Reprinted from the May, 1986 issue of CURRENT NOTES:
The Newsletter for Atari Owners.

From Prototype to Product

In this column and, perhaps, elsewhere you have read announcements about a variety of new Atari products. But, somehow, after you hear of the product nothing seems to happen for months at a time. Where are these new marvels? Somewhere on that journey that takes an idea and transforms it into a commercial product. Let's take a moment this month to review what is actually involved.

Atari is now primarily a hardware company. The first step in creating new hardware is engineering. The product must be designed or engineered. When the various hardware components are put together to make a model, software must be created to work with that hardware. In most instances the software development is much more time consuming and expensive than the hardware development. When hardware and software work together, a packaging team works on putting the product into some kind of attractive packaging that would appeal to consumers, be inexpensive, and still meet all functional requirements. Now the product is shown to Jack Tramiel.

When the product is given the go ahead, the prototype is shipped to the Federal Communications Commission (FCC) for approval. Actually, two prototypes are constructed and the second is sent off to the Underwriter Labs for the UL stamp of approval. Because of their normal backlog, anything sent to the FCC sits around for three months before anyone can even start working on it. While waiting for FCC approval, the factory can start making plans for mass producing the product.

If the FCC and UL approve the product, factory production can start; if not, it is back to the design phase to try and correct whatever faults were detected. Building a prototype by hand is a very different process than trying to build the same thing on an assembly line. When the first units start rolling off the assembly line, they have to be tested to see if they still do what they're supposed to do. If they fail the test, back to the factory for reworking. If they pass, then the FCC and UL are once more called in. You see not only the prototype but the final production model must have FCC approval. If the product passes these final tests, it gets that long sought after certification sticker.

Done? Nope. Now the production line swings into full gear, products are produced (in Taiwan) and must be shipped to the US. Once in the US, Customs steps in and does its thing. When the product passes customs, Atari quality control takes over. On any new product, one out of every two units are tested. This whole production, shipping, customs, testing process takes a minimum of about six weeks.

Another two to three weeks will still be needed to get the product in the distribution pipeline from Atari, through distributors, to, finally, your local stores. There, provided there is no long waiting list, you can now go and purchase the product.

New Product News

Now that you have some idea of the production process, let's see where things stand. The release date (when the product should be available in local stores) for the 20-MB hard disk drive is May 26. The first two prototypes of this product received their FCC approval in early April. The final version will have a metal case (just like the drives shipped to developers). The drive is a half-height Seagate with an Adapttec controller. The initial price will be $799, however, Atari is planning on producing their own controller for the drive and when they do the price will come down an additional $100.

The IBM V20-Emulator was sent to UL and to the FCC for approval at the end of April. This means we will probably not know until late summer whether FCC approval is forthcoming. By the way, the name derives from the NEC V20 chip (an 8088 clone) used in producing the emulator. While we are on the topic of emulators, you should be aware that emulating an IBM on an ST will not be your only option. Atari WILL BE producing an ST Emulator for the IBM. Yes, IBM owners will be able to run ST software too.

Look for a 1200 baud ATARI modem in June. It will be 100% Hayes compatible -- it will even look like a Hayes except for the case being gray. Anticipated price is $100. The modem will have both an RS-232 port as well as a serial I/O port so it can be used by either ST owners or XE owners. For an additional $20, you can get the necessary cables and software for the modem. Atari has developed software for the ST and the XE as well as for Apple and Commodore computers. Although no firm decision has been made, Atari is taking a look at producing a 2400 baud modem since the incremental cost would only be about $50-$60.

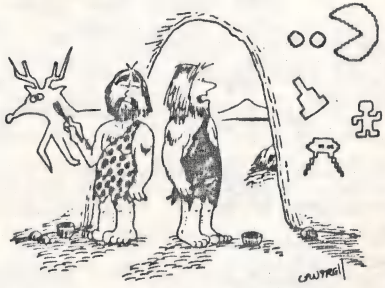
At the moment it looks like Toshiba will get the nod on the Atari Laser Printer. While negotiations are still underway, Atari is working internally on producing a desk-top publishing program to go hand-in-hand with their laser printer. (Hey, Jack! We may be able to make this an ALL-ATARI publication afterall. Just let me know when you want some testing done.)

DRI is no longer involved in future enhancements to TOS; the work is all being done internally at Atari. Efforts are centering on fixing known bugs, allowing use of a 60MB hard drive, developing a loading routine so you can boot from the hard drive, and allowing the addition of more (up to 21?) desk accessories. Current TOS chips are now widely available. Indeed, many more TOS chips have been shipped than ST computers. Hmmmm. Where are all those TOS chips going? Is anybody making an ST cartridge for the MAC?

The blitter chip is still being developed. Latest production prototypes have shown a tendency to blow up the ST. Whoops. Back to the factory. When things are working correctly and the chip is available (perhaps 2-3 months), it will be sold, complete with a new set of TOS chips, for $80. Final (as of this writing) decision was NOT to provide a socket in the 1040 for a blitter chip. 1040s will be upgraded just like the 520s, by installing a daughter board.

In the Black

In a recent interview shown on "Strictly Business," Jack Tramiel announced that Atari was in the black for first quarter 1986 to the tune of $9 million. He also reiterated his philosphy of providing a quality product at an inexpensive price and his intention to sell computers to the masses. Jack expects that Atari will have 20 percent of the personal computer market by 1990. Do you belive him? If so, save your nickles and dimes because Atari will be going public in the not too distant future. By the way, how's your club treasury. In a talk given at the Jersey Atari Computer Society, Sig Hartman said that Atari would provide user groups with a 1040 ST at distributor cost. Well, now, we might be interested in that Sig. Tell us more.



"GEE, MY WIFE DOESN'T LIKE THEM, EITHER. MAYBE I'M JUST AHEAD OF MY TIME."

Reprinted From Pokey Press

Atari ST Tips and Tricks
by Gordon Billingsley

BASIC:

ASTUN Volume 1.3 reported using input$ in place of inkey$ in ST BASIC. The syntax is a$=input$(1), with a$ storing whatever key is pressed. Changing the number in parentheses changes the number of key presses the program waits for. a$=input$(3), for instance, waits for you to press three keys before continuing with the next program line.

There is a problem with input$ in my version of BASIC (don't know if there were later releases to fix this problem). When a key is pressed the screen goes through an annoying, time-consuming convolution that involves closing the output window, opening the command window to test for the key press, closing the command window, re-opening the output window, refreshing the output window with the new information and then continuing the program.

I've discovered this annoyance can be by-passed by RUNning the program while the edit window is open. That can be done by typing RUN on a line by itself and hitting return or by using the mouse in the drop-down menu. I'm getting rather used to this little rolling rodent and prefer the mouse. It actually results in less movement and fewer strokes.

The now well-documented VDISYS() functions, which give lightning fast access to complex graphic and text functions, can accept parameters from easily written, one-line BASIC statements.

The BAR function, for instance, will respond to--and create different types of bars on the screen in relation to--parameters for fill interior style, fill style index, fill color and others. Using VDISYS(1), the program lines to set, say, the fill style indux would be.

```
10 POKE CONTRL,24
20 POKE CONTROL+2,0
30 POKE CONTROL+6,1
40 POKE INTIN,index:'where index=fill index number
50 VDISYS(1)
```

All of that--and similar sequences for four other parameters--could be replaced by the smaller, easier to writh one-line BASIC statement:

```
10 COLOR [text color, fill color, line color, style, index]
```

I used the technique while writing a record-keeping program that emulated check writing. I used the COLOR statement to set parameters for a pattern that approximated the background pattern on checks and then used the BAR function with a VDISYS() call to create the visual illusion of a check on the screen, all within the blink of an eye.

DEGAS:

An early version of DEGAS (Design and Entertainment Graphic Arts System) from Batteries Included will turn monochrome pics reverse (like a film negative) once they are recalled after being saved.

Batteries Included gave me these two fixes:

1 - Before running the program, set the pallet to reverse in the desktop control panel. This did not work for me, and it would be a hassle if you wanted to create as well as recall pics in a single session.

2 - They have fixed the bug and will provide an upgrade for $10.00 and the origional disk.

I also discovered two things about printing monochrome pics.

1 - If the printer drivers in DEGAS don't support your printer just use the screen dump feature [ALTERNATE] + [HELP]. The screen does not show any extra characters (such as GEM window) and therefore, you get a printout of your pic and nothing else.

2 - If your DEGAS version is turning your monochrome pics reverse, take heart, they don't print that way. No matter how funny they look on the screen, they come out positive on the printer.

1ST WORD:

This GEM-based word processor, which was given away by Atari with some ST purchases, also is sold at a reasonable price. It's been getting good reviews in almost every Atari magazine. It has one feature that makes it useful for playing around: It recognizes and will edit ASCII files. That means you can use it as a handy editor for programing and it also allows you to edit system files on the ST.

I used 1st Word, for instance, to edit the DESKTOP.INF file. I changed the name of the TRASH icon to SHREDDER. I was going to change to INCINERATOR as the ST manual suggested was the nature of the TRASH can, but then I realized thet using now commonly available disk utilities to recover deleted documents was a little like pasting scredded documents back together.

I also used it to edit the printer driver configuration files that come bundled with it. My Panasonic KX-P1091 is Epson compatible, so I used the LX-80 file and edited the printer name line to be KX-P1091. Now, when I print the little printer window in the upper left corner shows my printer's name and not Epson.

Further, I was able to suppress the annoying form feed that procedes every print job. The line in the table that reads:

```
20, C      *Vertical initialization
```

was edited to make the "C" a "0" (zero). It no longer wastes a sheet of paper every time I print something.

I've found it useful to keep two printer drivers ready at all times--one for form feed paper and one for page-at-a-time feed paper. I edited the printer file, which assumes form feed, to make it a page-wait type and saved it under the name PAGE.HEX. The origional form feed driver was saved as FORM.HEX. Now if I want to switch to single-sheet feeding, I need only run INSTALL.PRG before I run the word processor and select PAGE.HEX from the selector box.

To edit the driver, I need only load it into 1st Word and carefully read the comments to identify where the setting for page-wait or form feed was located (it's very near the top of the file). It is a toggle setting: 0=form and 1=page.

Clicking the mouse arrow in the shaded portion of the slide bars will scroll the page one screen (horizontally or vertically depending upon which slide bar you click.) This is very hand for reading a document on the screen. Clicking on the arrows scrolls vertically one line at a time and horizontally five characters (the length of the average word) at a time.

MEGAROIDS:

Just in case anyone hasn't found the undocumented use for the F1 key: Pressing F1 resets the program by ending the current gane and starting another game (if the last game is not being played). My cat showed me this trick once while watching me play from his front row seat in my lap. I didn't ask him where he learned it.

ADDITIONAL TIPS
By Robert Mann

I received the 97K ramdisk program and compiled it. It works great but I wanted to have two ramdisks available once I received the TOS ROMs. I went over the listing and poured over the documentats from the Developer's kit. By changing a few lines I was able to install a second ramdisk on the ST. All you have to do is change all occurences of "dev !=3" to "dev !=4" and in main instead of "¦ 0x8L" make it "¦ 0x10L". This last change properly sets Drvmap bits for 2 disks and 2 ramdisks. After recompiling and linking with accstart, go back to the desktop and install 'E' ramdisk, save the desktop and reboot. You will then have two 97K ramdisks to use in compiling programs.

If you edit the desktop image with a regular WP program or STWRITER and then print it to disk, you can make TOS recognize other file extensions with a double click of the mouse. The ROM O/S recognizes *.PRG, *.TOS, *.TTP, and *.APP (not in the documentation that I've seen). I edited the file and now my ST will run *.EXE applications from the desktop. The line to insert is

```
#G  03 FF  *.EXE@ @
```

A BASIC TIP
by Brian Duggan

You can easily write BASIC programs using STWRITER by following this procedure:

1 - Each line must start with a left edge flush line number.
2 - The BASIC line starts at the seventh column of the line.
3 - No more than 80 columns per basic line (alas!).
4 - Each line must end with a carrage return.

Once the program is input do the following reformat:

1 - Wipe out the printer controls at the head of the BASIC program (if they exist) and replace them with this line.
(Control L),0,(space),(Control R),3,0,0,(space),
(Control T),0,(space),(Control B),0,(space),(Carrage Return)
(The two lines are entered as a single line)
2 - Put the cursor in the first line, then enter the following.
(Shift Function 7),(Carrage Return),(ESC) followed by
(Control 0,1,0,(space),(Carrage Return),(ESC)
3 - After reformat is completed print the file using the print to disk option.

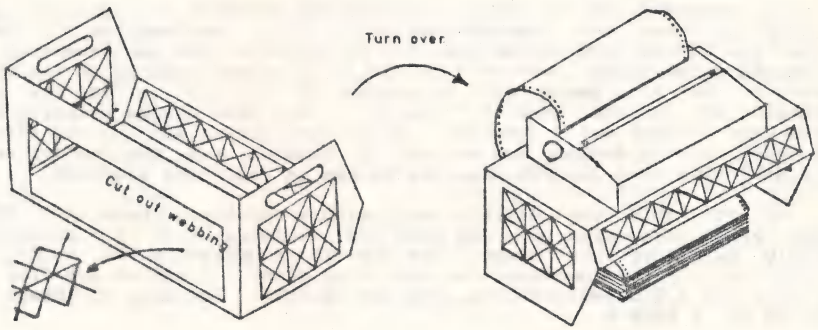
You now have a loadable BASIC program. Enjoy!

ASTUN Feels tips such as these will be helpful to many users. If you know of any techniques that will be useful to others please pass it on to us. If we publish it your subscription will be extended one issue free of charge.

The Computer Handyman
by J. Pazienza

A Reprint from The Pokey Press of The Atari Computer Club of The Palm Beaches

Does your stack of printer feed paper twist it's way thru a maze to get to your printer? Does it require the dexterity of a carnival rubber man to change your paper? Well your troubles will be a thing of the past after you've constructed this simple and inexpensive front loading printer stand. Simply purchase a 7 1/16 x 14 x 9 1/4 "Storagemate" modular plastic storage bin and cut out the back side webbing as shown. Turn it upside down and presto an instant front loading printer stand for 2 to 4 dollars. The same unit can also be used as a low cost monitor stand for a 13 inch or smaller TV depending on the weight.



Excerts from ATARI USERS GROUP NEWS (A publication on Atari Corporation)

BYTE MAGAZINE HAILS 1040ST

In the March 1986 issue if Byte Magazine, Editor-in-Chief Phil Lemmons said, "The Atari 1040ST is one of the great milestones in personal computing." In Byte's product preview (which was that issue's cover story) they had lots to say to expand on this. We have extracted some interesting material below:

"Atari's new $999 1-megabyte 1040 ST establishes a price break reminiscent of the Commodore 64. And the 1040 ST begins retail life at a price which represents less than one dollar per K-Byte. The 1040 ST is a clear bargain, with over 1 megabyte of RAM, its operating system in ROM, an internal 720K byte double sided drive, an internal power supply, and the same features and functionality that already make the Atari 520 ST an attractive purchase."

"There are a number of changes which are more than cosmetic. The internal power supply eliminates two of the external power supplies needed by the 520 ST (wire haters rejoice). We left the unit on for five days and experienced no difficulties with overheating. There is no internal fan, but the unit appears to provide adequately for the dissipation of heat. The internal disk drive supports both single and double sided disks."

"The 1040ST has a remarkable price, and it shall rest for some time as a clear leader in price-performance. The power supply is internal, and TOS is in ROM. And given the outstanding price-break, our overall impression is even more positive."

Accompanying the Byte article is a table (on page 87) which shows the 1040ST as the clear winner in value, overtaking the previous leader which was the 520ST.

The article did have some misinformation, since it was written last December and some specifications changed. There is no Rf modulator in the 1040ST, although this has appeared in the latest shipments of 520ST computers. Also, there is no graphics coprocessor planned for the 1040ST -- this chip will be worked into a future model in the ST line at a higher price, and will be available only as an upgrade board for existing ST computers.

COMDEX BOOTH PREVIEW

The Atari booth at Comdex features the ST family of products and includes over 30 software development companies.

Developers participating include Batteries Included, Michtron, Haba, Hippo, Activision, Audio Light, Mark Williams, MaxThink, Migraph, Mirage Concepts, OSS, Philon, Progressive Computer Applications, Quickview Systems, Spinnaker, Sierra On-Line, TDI, Unison World, VIP Technologies, VersaSoft, Avila, XLent, First Byte, Abacus, Power Systems, and QMI.

Third party products being introduced at Comdex include the acclaimed accounting program Dac Easy from Dac Software Inc. (214-458-0038) which was chosen as Software Product of the Year by InfoWorld, MaxThink's (415-428-0104) idea processor, a laser printer interface from Pacific Rim Data Sciences (408-272-5665), One Write accounting from Sierra On-Line, a speech synthesizer from First Byte (213-595-7006), PC-CAD from Abacus (616-241-5510) for engineering design, and a local area network from QMI (315-451-7747)

Especially for Comdex, Atari developed a 400 page book describing three hundred software titles made for the ST system, more than half of which are out now with the remainder available during the next quarter. This book will be available after the show for $9.95 plus $2.50 for shipping and handling, from Atari Customer Relations, 1196 Borregas Ave., Sunnyvale, CA 94086, Attn: ST Software Book.

New products being shown by Atari at Comdex include the hard disk (see below), the SMM804 printer (selling for $219.95 for an 80 column, 80 cps printer with full ST compatibility), a preliminary MSDOS emulator in hardware, and Atari's own brand of 3.5" floppy disks (certified double sided, part number DS5031, wholesale cost of $10 per box of 5).

The SMM804 printer is currently in stock in the warehouse and available for immediate shipping, as are the floppy diskettes.

New Atari software packages at Comdex include the CP/M Emulator, the Atari Planetarium (a wonderful astronomy package that turns the computer into a planetarium with stars, planets, galaxies, phases of the moon, eclipses, and

Clip Elite -- and the Isgur Portfolio Series. [Mirage Concepts] offers H & D Base, while Versasoft puts out Db Man -- [a dBase II and a] dBase III clone. Sierra On-Line plans to introduce an accounting package, [ST] One Write, in the next few months."

"John Sobernheim, owner of The Computer Room, Aurora, Colo., a five-year-old Atari dealer, sold out his first shipment of 12 1040ST's within days."

NO HOLDS BARRED COMPARISON: ST VS. HIGH END 8-BIT COMPUTERS

It amazes us that some people will still pay $500 to $1000 for an 8-bit computer system, when for the same price they can get a complete 520ST setup that will outperform any 8-bit machine in every measurable detail.

There are two good examples of overpriced 8-bit systems that are still available in today's marketplace: the Apple II series and the Commodore 128. Complete with disk drive, the Commodore is less that $100 below the price of a 520ST, while the Apple actually costs much more -- for a more poorly performing product. How can this be?

Inertia.

Both products are coasting on the existing base of their predecessors. The original Apple II was first sold in 1977, while the C128 is based on both the C64 (1982) and CP/M (circa 1976). The advantage is that both systems are somewhat compatible with a large software base, and are known entities for institutions like schools. But is this really a good thing?

Both machines are designed around what is essentially outmoded technology. Lets face it, the semiconductor field has advanced quite a bit every year since these machines were developed. Chips can now integrate many more circuits than they could a few years back -- one of the main reasons why Atari can make the ST system for so little compared to these "toy" computers. At this time it costs less to design a system around the powerful 68000 chip than it did to build around the simple 6502 or Z80 series a few years back.

MICROPROCESSOR: the ST uses a 68000 chip which handles data 32 bits at a time internally and 16 bits at a time externally. The 6502 and Z80 are limited primarily to 8 bits, so they have to work harder on big operations. Above all this, the 68000 works at 8 million cycles per second compared to 1 or 2 million for the 8-bitters. Not only does it work harder, it does the hard work faster.

BUILT-IN SOFTWARE: the operating systems of the Apple and the Commodore are primitive 16K programs compared to the 192K of TOS in the ST. Where the 8-bit systems provide basic input/output functions with a separate DOS, the ST integrates both of these along with graphics features, windowing, menus, and mouse, plus the desktop. The additional programming in the ST makes it easier for users to understand.

MEMORY: the 520ST includes 512K of RAM, where the Apple and the Commodore only provide up to 128K. The ST has big-league capacity for the kinds of problems that you really need a computer for.

VIDEO DISPLAY: the ST gives you 3 modes to choose from, including a special high-resolution monochrome mode of 640x400 pixels, perfect for text operations. Both the ST's monitors provide unusually crisp and vivid images that the 8-bit machines can't compete with. This is because the "video shifter" chip in the ST uses the latest in technology and operates at an astounding 32 million cycles per second.

SOFTWARE: here's where the public percieves an advantage for the 8-bitters. They have thousands of packages to choose from today, compared to hundreds on the ST. But numbers alone don't tell the whole story. The kind of software we are seeing on the ST is far beyond the capabilities of the 8-bit systems. Word processors use the ST's window environment to provide ease of use that is impossible on the 8-bits -- they don't have the speed or the memory capacity to make it practical. Sophisticated personal and business applications abound on the ST that are completely impossible on the 8-bits. And the rate at which new titles are released for the ST has been far greater than for the 8-bits, so the disparity in the number of titles will disappear very soon.

So, when someone suggests that one of the high-priced 8-bit computers are anything less than relics of a previous age, we just shake our heads and wonder.

even Halley's Comet), Star Raiders ST (the one everybody's been waiting for), VT100 emulator (terminal software on cartridge), and DB Master One now in packaging for general sale.

ATARI HARD DISK LAUNCHED AT COMDEX

Atari is introducing the SH204, a 20 megabyte hard disk drive which will retail for $799 and is due to ship toward the end of May. Atari expects hard disks to be as indispendible for the ST as floppies were on the last generation of PCs.

With a hard disk, a 1040ST system will retail for less than $1800, compared to about $3000 for a comparably equipped Amiga and well over that for a Macintosh or an IBM PCAT. Once again, Atari provides Power Without The Price.

WEST COAST COMPUTER FAIRE REPORT

Atari stole the show at the West Coast Computer Faire without even having our own booth. Two user groups formed the nucleus of ST activity: ABACUS from San Francisco, and the San Leandro Computer Club. The latter group drew huge crowds with demonstrations of the Mac Cartridge, a full Macintosh emulator for the ST. See some of Jerry Pournelle's comments below for more on this.

Among the Atari-related events at the show was a panel discussion including Atari president Sam Tramiel and software development vice president Leonard Tramiel. This was a standing-room-only event which was attended by the press and many local Atari enthusiasts.

One of the most popular products from Atari was Star Raiders ST, which was displayed at the booths and on a wide screen TV at the ABACUS meeting. Even in an unfinished form, the demand was incredible.

JERRY POURNELLE AT THE WEST COAST COMPUTER FAIRE

In Jerry Pournelle's column in InfoWorld for April 14, 1986, the widely read columnist reported these impressions:

"Most of the excitement was centered around the Atari ST and the Amiga machines. As usual, there were a lot more Atari users than Amiga people. I would estimate I saw at least five times as much practical software for the Atari ST as for the Amiga."

"Atari was there in force. Sam Tramiel, president of Atari, and his brother Leonard, vice president for software development and a dyed-in-the-wool hacker, held a combination press conference and user group meeting that was one of the best events of the show."

"I don't know Atari's exact sales, but it's clear to me they've sold a lot of 520s -- and although sales of 8-bit Atari machines are flat, they're still high. ...from here on, you had better think of Atari as a major player in the computer game."

"The most exciting product at the Faire wasn't for sale. Data Pacific Inc. demonstrated the Mac Cartridge, a device that plugs into the Atari ST and turns it into a Macintosh. Now you can get an Atari 520ST with monochrome screen, a full-featured, dot-matrix printer, the Mac Cartridge, and Excel for under $1,500. As a bonus, Excel runs 20 percent faster on the ST with Mac Cartridge, and the ST has a screen large enough to see."

"As soon as I file this column I'm going over and do my expense report. I'll use Excel on the Atari ST."

COMPUTER + SOFTWARE NEWS TALKS TO ST DEALERS

In the April 14, 1986 issue, Computer + Software News, a publication not noted for its friendliness toward new products, talked to some of our dealers and had this to report:

"Atari is landing in businesses as a substitute for a dumb terminal, says Vel Selvan, store manager of Access to Software, San Francisco, who reportedly sells 10 times more Ataris than Amigas."

"A lot of corporate sales are happening with the 1040," says Selvan. "It's a frenzy."

"Atari software publishers also have a cache of business titles. Batteries Included says it is working on an advanced word processing program -- Paper

dBMAN from VersaSoft is one of the best database management systems we have seen. It is almost entirely compatible with Ashton-Tate's best-selling package "dBase III". Atari Corp. uses Db Man in our customer relations and service departments to automate our order processing and invoicing system. Dealers like San Jose Computer have reported that their operations are up and running on dBMAN as well.

dBMAN is not copy protected, and can be used with ease on a hard disk drive. It supports up to 2 billion records per file, 10 active database files (as opposed to 1 on dBase III), parent-child relationships of records, easy menu creation, and a powerful debugger/editor.

dBMAN is available from Atari Corp., 1196 Borregas Ave., Sunnyvale, CA 94086, phone (408) 745-2000.

Easy Draw from Migraph is a very impressive drawing program for computer graphics, page design, and a host of other uses. Retailing for $149.95, this package supports all three of the ST's resolution modes and any Epson FX-80 or compatible printer.

This is a very different program from the "paint" programs previously available. It is object oriented. Rather than manipulating individual pixels on the screen, you work with geometric shapes that can be changed after they are placed down. Individual objects can be worked with separately, or you can put them together into groups which are then acted upon together for as long as you like.

Easy Draw is not a simple package to get to know. Where a "paint" program can be learned in a very short time, Easy Draw is powerful and subtle enough to offer more features each time you use it. As such, many people will find this harder than they expect in the beginning -- but patience is amply rewarded by top-notch results.

Easy Draw is from Migraph, Inc., 720 S. 333rd (201), Federal Way, WA 98003, or phone (206) 838-4677.

Solapak is a package that doesn't look like much but quickly becomes indispensible. It includes a trio of products: a RAMdisk utility, a print spooler, and a crt saver. What makes Solapak stand out is the extra features programmed into them.

The RAMdisk is automatic -- when placed on the disk used to boot up the system, it sets up automatically each time you use the computer. There is a file on the disk which can be edited (using ST Writer or most other editors) to customize the size of memory allocated. In our 1040ST, we set up a 512K RAMdisk.

The print spooler is much more than just a spooler. A normal spooler lets you print one file while performing any other function on the computer -- a limited multitaskign function. The Solapak print spooler lets you print up to 10 files at a time. It also includes many formatting options, which can be specially customized for your printer. You can select type styles, margins, and any other printer feature from a handy mouse-driven menu at the time you start the spooler going. There are many files on the Solapak disk with the setups for most standard printers (including the Okidata Microline 193 we use for this newsletter). It is also easy to edit these files for different models.

The crt saves would be most useful for people with TV sets on their ST's and for those who leave their machines running unattended for long periods. Very simply, it turns off the screen when the keyboard hasn't been touched for a while. For those of us who don't want this active, it is simple to turn off this function.

Solapak is available from Solar Powered Software, 1807 N. Evergreen, Chandler, AZ 85224.

PrintMaster is similar to the popular "Print Shop" package from Broderbund. It customizes printouts for signs, banners, calendars, stationery, cards, flyers, etc. Many different options are included like border designs, type fonts and styles, and layout patterns, all selected from menus.

Print Master supports the most popular printer models including most Epson, Okidata, and Star Micronics models.

A separate package called "PrintMaster Art Gallery I" is a supplement which includes over a hundred extra pictures for use with PrintMaster. There are zodiac signs, sports figures, animals, office items, and much more.

PrintMaster and the PrintMaster Art Gallery I are sold by Unison World, 2150 Shattuck Avenue, Suite 902, Berkeley, CA 94704.

APPLICATION NOTE: THE ST AS A BBS

We have talked about the ST's use as a terminal in prior issues of this newsletter. Another easy project for an ST system is to turn it into a dedicated bulletin board system (bbs).

The best system configuration for a bbs is a 520ST with any monitor, plus one floppy drive and an Atari hard disk. You don't really need the extra memory of a 1040ST for a bbs -- the current batch of bbs software doesn't use the extra RAM anyway.

At Atari, we have had our own bbs running with this setup for months now. During that time there have been no hardware problems at all. In fact, we have 4 complete ST systems, each with its own phone line and modem, taking about 2000 calls each week.

Since Atari is not making modems for the ST yet, you have to make a choice here. The best bet is a "dumb" modem than handles 1200 and 300 baud. The auto answer feature is essential. Beyond that, extra features get in the way. Clever users can figure out how to mess up a "smart" modem if they try -- it's best not to give them the opportunity.

On the Atari bbs we use Racal-Vadic model VA3451 and VA3212 modems, which appear to be identical -- they are fairly old and were lying around in a back room before we set up the bbs, so we don't know much about their history. They work fairly well, although they occasionally leave the phone off the hook and have to be manually reset (a pain when this happens on Friday night and stays off hook until Monday).

Now to the issue of software. Except for hardcore hackers, who can write their own, most people opt for an off-the-shelf bbs package. There are two in popular use on ST's today: the Michtron BBS and FoReM ST. We use the Michtron BBS, partly because it was out first and also because it is easy to use, both for the system operator (sysop) and for the new caller. FoReM is much more powerful, but requires lots of initial setup work and demands more training of callers.

There are two more packages we know of that are being completed. Switchboard, from SST (the Chat! people), sounds like it will have some good features, especially in the message area.

NiteLite from Paul Swanson is an upgrade from an 8-bit Atari BBS (as is FoReM), which Atari used to use on the bbs before we went to ST's. NiteLite is almost as easy to set up as the Michtron BBS, but is more easily customized to your needs as time goes by. We are waiting for a copy of this to see how well the ST version performs.

Speaking of 8-bit bbs's, there are many out there. We upgraded for a few good reasons. One, we liked the 20 megabytes of hard disk storage -- we now have about 5 megabytes of download files and nearly a meg of messages online. The ST is also blindingly fast, especially with a hard disk -- much faster than the big systems like CompuServe.

The Atari bbs is called "Atari Base", and can be reached by calling (408) 745-5308. Special interest groups (SIGs) on the bbs include; 1.) General, which includes information of general interest and is open to anyone, 2.) XE, which contains data of interest to 8-bit users and is available to all, 3.) User Groups, a private sig, available to user group officers, 4.) Dealers, which is only open to dealers, 5.) Service Centers, which is for use by service centers only 6.) Developers, a private sig for developers 7.) ST Computers, and 8.) ST Art, both are open to anyone, and 9.) Sysops, a private sig for sysops.

The Michtron BBS is available from Michtron, 576 S. Telegraph, Pontiac, MI 48052, phone (313) 334-5700.

FoReM ST is available from Comnet, 7348 Green Oak Terrace, Lanham, MD 20706, phone (301) 552-2517.

Switchboard will be available from SST Systems, 3456 Willis Dr., Box 2315, Titusville, FL 32781, phone (305) 269-0063.

NiteLite ST will be available from Paul Swanson, PO Box R, Billerica, MA 01821, phone (617) 663-4463.

We received three new software packages from MichTron. A description of these programs follows. As we learn more about them we may follow with a full review in a coming issue.

DFT IBM to Atari ST Direct File Transfer

First we received DFT a program that allows the user to transfer files between the Atari ST and any IBM compatible computer. The package contains two disks. One for the Atari ST and the other for the IBM.

In order to use this program you must obtain or build a standard null modem cable. Instructions on making the cable are contained in the manual. If you can not obtain one locally or make your own then MichTron sells a ten foot one for $39.95. The cable is then hooked up to the serial port on the Atari and a serial port on the IBM. Any IBM serial port may be used.

You then load the approiate disk in each computer and boot the DFT program. All commands are typed into the ST computer. You can display the directory for IBM as well as the Atari. You can then transfer files in either direction. You can also change directories on either system.

If you own both an IBM and an Atari ST this program should prove valuable. Even if you don't have an IBM but you have a friend that does this may be just what you need to obtain data files and various templates. Such as Lotas 123 applications for use with VIP on the Atari.

KISSED A Self Relocating Debugger

This program allows full screen editing. It displays registers and blocks of memory. It allows multiple pass breakpoints. It includes a miniassembler and disassembler.

This program may not be useful to a casual ST user. However if you do some serious program development, then this tool may be a very useful tool worth considering.

DOS SHELL A Command Line Interpeter

If the ST is not your first computer, you may feel more comfortable with a command interpeter than always having to use the mouse enviroment provided by GEM.

DOS Shell is a program that mimics the MS.DOS enviroment so popular on the IBM P.C. It provides all Major Elements of the MS-DOS Shell.

A person familiar with MS DOS will be right at home with this program.

As a shell it is more than just a simple interpeter that executes commands by typing its name. It allows full use of wildcards, redirection of output, pipes and filters.

By combining these features with the MS DOS commands you can actully execute programs from simple batch files to complex processes.

To get a list of commands simply type help. Following is a list of most of the DOS Shell commands with a very brief description of each.

CD,CHDIR	Change Directory	CHDSK,CH	Check Disk
CLS	Clear Screen	COPY	Copy Files
DATE	Check or Enter Date	DEL	Delete Files
DIR	Show Directory	DEL	Delete Files
EXIT	Return to Desktop	ECHO	Provide Prompts
HELP	Dislplay Directory	IF	Comparison Statement
MKDIR,MD	Make Directory	PATH	Set Directory Path
PAUSE	Suspend Execution	PROMPT	Set System Prompt
REMARK	Enter Comments	RENAME,REN	Rename Files
RMDIR,RD	Remove Directory	RUN	Execute Program
SET	Set Environment	SHIFT	Shift Command Paramiters
TIME	Set or Read Time	TREE	List Subdirectory Paths
TYPE	List Fuile Contents	VER	Get Version Number
VERIFY	Verify Disk Saves	VOL	Show Disk Volume Name

This Program should be considered If you are used to, or wish to learn the MS DOS Environment.

NEW for the ST . . . from MichTron

UTILITIES made for speed, efficiency, and simplicity.

KISSED *Debugger by Keith Enge*
Announcing the ultimate programmer's tool: this complete debugger features full screen editing, tracing and execution options, "timed" breakpoints, independant screens, dynamic relocation, assembler-disassembler, decimal/hex conversions, over 40 commands, a help key and more!
For the Atari St $39.95

DOS SHELL *Utility by Timothy Purves*
DOS Shell makes your ST mimic the MS-DOS command structure. All the familiar MS-DOS cammands are available from GEM. Add the extra power of "global" file commands, multiple-file manipulation, and batch files to your Atari.
For the Atari St $39.95

CORNERMAN *Utility by J. Weaver Jr.*
Clear off your cluttered desk! This single utility gives you an electronic notepad, calculator, address/phone book, phone dialer, ASCII chart, clock, and even a "15 squares" game. Install *Cornerman* as a desk-top accessory, and it becomes available almost everywhere!
For the Atari St $49.95

ECHO *Environment Controller by Timothy Purves*
This software/hardware combination lets your ST control electric appliances. *ECHO* regulates thermostats, dims incandescent lights, and has a timer program that even takes weekends and holidays into account. It's perfect for security and energy conservation. *ECHO* uses inexpensive X-10 remote plug-in modules, so it's completely wireless!
For the Atari St $39.95

D.F.T. *Transfer Utility by Timothy Purves*
Transfer files between your Atari ST and IBM computers. With this program and your own modem or direct-connect cable, you can convert all your important ASCII data-files quickly and reliably, without retyping.
For the Atari St $49.95

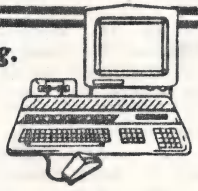
THE ANIMATOR *Graphics Utility by Keith Enge*
Now you can animate pictures made with Degas or Neo. Add flair to business presentations or make your own movies for fun!
For the Atari ST (Degas and Neo not included) $39.95

PERSONAL MONEY MANAGER by John Kring
This specialized database keeps track of personal finances, checking records, and supplies a variety of well organized reports.
For the Atari St $49.95

Last month's features:

Business Tools	Over 80 Business Forms	$39.95
MichTron UTILITIES		$59.95
M-DISK	RAM-Disk Emulator	$39.95
MI-TERM	Communications Program	$49.95
SOFT SPOOL	Printer Utility	$39.95
CALENDAR	Desktop Appointment Calendar	$29.95
MI-DUPE	Fast and easy file duplication	$29.95
BBS	Complete Bulletin Board System	$49.95

GAMES designed to be fast, colorful, and exciting.



TIME BANDIT *Arcade Game by Dunlevy & Lafleur*
Explore medieval dungeons, western frontiers, and future worlds in one game! Each of 12 worlds has over 15 levels and is a game in itself! This fast-action arcade game even has a built in adventure: pilot the *Starship Excalibur* as you try to rescue its vanished crew. Great sound, beautiful graphics, and hundreds of screens: *the conquest of Time awaits!*
For the Atari St with color monitor $39.95

MAJOR MOTION *Arcade Game by MacKenzie & Sorenson*
Race down the highway in this exciting spy-chase arcade game. Enemy drivers, deadly helicopters, and gaping potholes threaten to destroy you. Defend yourself with smoke screens, machine guns, oil slicks, and missiles or escape down branching roads and treacherous rivers.
For the Atari St $39.95

GOLD RUNNER *Arcade game by Dave Dies*
As Commander of the Load-Runners, you must infiltrate underground mines in search of gold and adventure. Use wit and skill to escape with the loot. Over 50 screens with narrow paths, steep ladders, dangling ropes, and hidden traps will challenge your reflexes and test your logic.
For the Atari St with color monitor $39.95

SOLITARE *Strategy Game by J. Weaver Jr.*
Five classic card games: play Solitare, Klondike, or Poker Squares by yourself, or test your strategy against the computer's in Cribbage. And if friends want to play, it's four against the house in Blackjack! The rules are accurate and the graphics amazingly realistic. Take a refreshing break from arcade games, without missing any of the fun and excitement.
For the Atari St with color monitor $39.95

Last month's features:

MUDPIES	Arcade Game (requires color monitor)	$39.95
FLIP SIDE	Strategy game	$39.95
LANDS OF HAVOC	Arcade Game (requires joystick)	$19.95

All reasonably priced, with more coming every day. Ask for our latest catalog!

Dealer inquiries welcome • Visa and Mastercard accepted • Add $3.00 shipping and handling to each order

MichTron

576 S. Telegraph, Pontiac, MI 48053
Orders and Information (313) 334-5700

Review of:

The Music Studio, (c) 1986 by Activision, Inc.
(developed by Audio Light, Inc.) List price $39.95
by Br. Tobias S. Haller, BSG

Overall rating: * * * * (out of possible 5)

"The Music Studio" is the first widely available music composition program for the Atari ST series computers. It's capabilities, though somewhat limited, are astounding considering the price.

DOCUMENTATION:
The program is on a single disk, copy-protected. It runs without TOS in ROM, and it seems that having TOS in ROM will not increase the memory capacity of the program, which is listed as 8,000 notes. The system requires a color monitor.

The instruction manual is pocket-sized, clearly written and complete. The order in which subjects are covered is somewhat confusing, but all you need to know is in there somewhere. The program itself is fairly intuitive, and the manual serves to clarify the few mysteries that come up.

Assorted sample compositions are on the program disk, which will give a good idea of the program / system capabilities.

The documentation covering the MIDI channels is specifically geared towards the Casio CZ-101 series synthesizers. As the equivalent "power without the price" in the synthesizer world, this makes a good deal of sense. It should not be difficult for the owner of other MIDI equipment to adapt, provided the equipment in question has good documentation.

OPERATION:
The program is loaded from the low-res disk menu, with a double-click on "AUDIO.PRG." The load time is reasonable, and the opening frame is attractive. The opening theme music is fun, and can be skipped once you've gotten past the whiz-bang stage and want to get down to work.

The opening work screen is an empty "grand staff" with a five octave range. The program is almost entirely mouse-capable. (One can use the keyboard instead, if desired.) Drop-down menus and options are the rule. Notes, rests, ties and measure bars are placed on the grand staff by selecting the option (it then turns pink), positioning the cursor and clicking on the staff. After a bit of practice one can zip about the screen. It is a trifle inconvenient when the note duration changes rapidly--one is constantly going to the duration setting box and then back to the grand staff--and this is where the function keys can come in handy. One wishes, however, that the function keys were labeled on screen.

Without a MIDI instrument attached, what you have is a fair three voice music program, and as such it isn't bad. When you add a CZ-101 or other MIDI instrument the result is impressive. With the "solo" switch (on the CZ-101) engaged, you can have four different instrument sounds at a time, out of a possible "palette" of fifteen. With the "solo" off, you can have one sound in chords up to eight notes deep (the maximum number of simultaneous notes on the CZ-101). By changing voices from the "palette" you can get a rich variety of sounds.

The only MIDI parameters that can be set from the program are the instrument and channel, the voice (preset, cartridge or internal) and the range. It is hoped that a subsequent update might include the ability to set the other parameters of the MIDI instrument from within the program. In addition, several MIDI capabilities are unavailable: volume (a serious weakness, meaning that the ability to accent notes is missing), pitch bend, and portamento.

The program's chief lack is in certain musical subtleties that render it less than ideal. The designers might address these specific problems in any contemplated update. The biggest problem is that several parameters must be set for the entire "song" or composition, and can not change during the course of the performance of the song (exept manually from the MIDI parameter screen). These include tempo (no rubato here, nor even a simple ritard; to say nothing of having more than one "movement" of a piece in one document file) and dynamics (farewell subito piano, and any effects such as crescendo or diminuendo -- pretty basic stuff in the music world!) These are serious flaws in what would otherwise be a much more sophisticated program.

What is provided, given these limitations, is excellent; and at the bargain price, this program is as much of a showstopper for musicians as "DEGAS" is for graphic artists. It certainly ranks as a must have for budding composers. It even provides, for the less than musically literate, a "painting music" mode, in which the grand staff becomes a canvas for placing blobs of color which can be translated into notes in traditional notation.

Disk access is fast, even long compositions saving and loading in seconds. One can also format (initialize) a disk from the program. Subdirectories (folders) don't seem to be supported.

Music can also be printed out on compatible printers. The quality of the printout is probably adequate for a record, but would be difficult to play or perform from.

OVERALL:
Though limited in certain respects, this program provides a good working tool for the amateur composer. A professional might use it for sketching out ideas for later realization. With the improvements suggested above, this program would be worthy of more serious consideration.

Hybrid Arts' DX Droid: The BEST MIDI software yet?
by Mike Seeley

Before I begin this review, I'd like to point out that I am not a hacker by any stretch of the imagination. Sure, I can poke around BASIC a little and I know how to smack my old Commodore 1541 on the side of the head when it needs it (frequently), but you guys who are looking for a complete dissection of this software's inner workings may be disappointed. However, if you're curious as to what a state-of-the-art machine like the ST can do with a serious MIDI program, read on!

Hybrid Arts' DX Droid is a MIDI editor and librarian for the Yamaha DX and TX series of synthesizers. For those of you who pay little or no attention to the music or electronic keyboard world, these keyboards are THE new standard in every band and recording studio in the world, and rightly so - they can sound like almost anything you want it too. The DX7 can be heard on practically every record you own. The only drawback is programming it. There are around 160 parameters to each sound (or "patch" as they're often called now). Sounds are creating by combining 6 sine-wave generators (called "operators") in 32 different ways (called algorithms). The sounds are shaped by envelopes for each operator and it all gets quite complex. In fact, programming the DX itself is like a nightmare, except more infuriating. Fortunately, Yamaha cut no corners with the DX7 - they made the parameter data available over MIDI, so you could edit and save your sound to floppy disk with any MIDI-card-equipped computer. This opened the floodgates, and now there are dozens of software-based patch editors and librarians out for the DX7 and the C64, 128, Apple II, Atari, and even (belatedly) the IBM PC.

MIDI (an acronym for music instrument digital interface) is a standard by which microprocessor-eqipped electronic keyboards can "talk" to each other over serial connections. Although originally intended only for playing one keyboard from another, MIDI has mushroomed as computer-based sequencers, etc. have become commonplace among musician's gear. Hybrid Arts has been the only company that I know of writing MIDI software for the Atari computers, and now they (finally) have beat everyone else to the ST marketplace with "DX Droid" -

a program with every conceivable kind of DX7 editing and filing function, AND an artificial intelligence-inspired random generation that can make up its own sounds with the click of a mouse button. Once the competition sees this they may just give up entirely! It's that good.

"DX Droid" loads on the ST quickly, and then prompts you for a formatted data disk. It then formats this disk to suit it's own filing system - apparently a non-standard type of file - which provides for 18 banks of 32 sounds PLUS 1032 "files", which is what "DX Droid" calls a patch (I wish they would have a standard name for these things!). That's a little over 1600 patches per single-sided disk! The speed at which DX Droid loads a bank of 32 sounds is AMAZING - about 3 seconds. Compare that to about 50-60 seconds on your basic Commodore! I was a little disappointed that "DX Droid" doesn't address the possibility of double-sided disks like mine. There's also no provision for using a second drive, which would be handy if you have zillions of sounds like me.

A minor annoyance - yes, "DX Droid" is copy-protected. It also will only load from drive "A", which means you can't run the program from a RAM disk. However, at least it won't crash a RAM disk you may already have installed - it just ignores it and runs normally. The program itself takes up about 110K, probably 200K total if you figure the various files & DX data you'll have in memory. Like many other programs, desktop accessories can't be reached while DX Droid is running. I didn't miss them though because I was having so much fun.....

There are 8 menu's that are used to access the main functions of DX Droid, and they're located logically on function keys 1-8. They are:

F1- The file menu (default screen-for single sounds)
F2- The edit menu (where all the great graphing is)
F3- The index menu (to keep track of all those sounds)
F4- The catalog menu (a sophisticated disk directory)
F5- The bank menu screen (for loading, unloading, etc.)
F6- The utility menu screen (the misc. screen)
F7- The edit bank screen (for the "workbank" in memory)
F8- The Droid menu (where the ST creates its own sounds!)

F9 is used to generate up to 16 MIDI note-on commands so you can "play" the DX from the ST's keyboard - a great idea. F10 is for quitting - DX Droid prompts you to make sure you want to quit before returining you to the desktop.

There's a LOT of "stuff" in this program. The edit menu when selected gives you your choice of sound editing, with either an on-screen display of each envelope's shape (very accurate too), a keyboard scaling edit mode that shows the keyboard on screen with the operator's up or down-ward scale above it (nice!), or a "view OP EG" (operator's envelope graphs) mode that can show graphs of up to all 6 operators in 4 different colors, so you can see EXACTLY what is going on in your sound (woops, I mean file). The output level screen shows a graph of each operator's output level. There is also the "standard" numbers-on-the-screen type of editing, but it's much better than most. Your "file's" algorithm is right there on the screen (no more looking back at that those little diagrams on the synth). You can copy or swap envelopes, mute 'em (very handy), and best of all use the mouse to scoot all over the place and change those parameters quickly. You get used to (and spoiled) by DX Droid's editor mode very fast, and before you know it you'll have lots of GREAT sounds.

DX Droid holds one "workspace" (equivalent to the DX7's "edit buffer") and one "workbank" of 32 soundfiles in it's memory. It does NOT hold all your patches in RAM like the rumor mill had suggested. Considering the speed of the disk storage, it won't matter. You can load in a bank, mix it or copy it, name it, print it, transfer to or receive from a synth, and it's all done as fast as you can say "8mhz operating system". In fact, in MIDI-receive mode you'd better be quick with the MIDI transmit button on your DX, because the DX Droid will only wait about 2 seconds before figuring you're a slow-moving cretin and say "synth i/o problem". And there's a even a quick loader for fast mass-transfer of program data to multiple and TX816 rack keyboard set-ups.

I would have liked to see more space on the disk for banks of sounds rather than separate files. There are so many file-slots that you could "lose" some sounds, but DX Droid programmer Tom Bajoras has wisely thought of that too. Each file can be searched as a word processor would do - by first

letter(s), last letter(s), or a combination of letters. Can't remember the name of that great sound, but you know it was "dog"-something? Well, DX Droid will find your "Dogfish_86" file for you! Each file can also be "indexed" - basically this creates a small filecard for the file which you can use for a description of up to 126 characters.

DX Droid Files can be "locked" and unlocked, individually or all-at-once. This prevents you from accidentally erasing your favorite sounds. DX Droid will even keep track of your revisions to a file. A real interesting touch is the "execute workbank" that is in the bank menu. Once you load in your bank and go to this screen, you can send any soundfile in the workbank to the DX with the touch of a single key (1-0 & A-V). This would be excellent for live keyboard use - perhaps the future may see ST's on MTV!

Now comes the fun: the Droid menu. Are you sick of starting sounds from scratch? Let the ST do it! There are two seperate Droid possibilities, one gives you a sound that has software-defined parameters which make it, supposedly, a more "use-able" sound. Droid-2 is a more random type of number generation. Frankly, I couldn't tell much difference between them, except that Droid 2 creations are crazier. What's incredible is that the Droid comes up with quite a few sounds that only need minor envelope or modulation editing (and hey! that's easy now!) to be good original sounds. And if pressing the button yields a clinker you can just press it again. This is REALLY fun. You can make whole banks of Droid-created sounds, even a whole disk at once if you want.

Besides generating it's own sound-parameters, the Droid function will "glide" from one sound to another in 16 steps. It's quite amusing to start with a train sound and step through the different parameter changes to an organ sound or whatever. Often some good sounds can be found midway. The "distort" function is yet another one-of-a-kind idea. When this is selected the Droid randomly changes parts of the sound, with each selection further from where you started (better save it first!). As you might have guessed, after 4 or 5 times the sound you started with is long gone, but who knows? Maybe you'll get something new and exciting. You can also find the "average" between two sounds, although this works best when used with two patches of the same algorithm.

I used DX Droid with both a DX7 and TX7 expander module, as well as with a Commodore SX64 computer with no problems. If you own a DX5 or a TX rack mount synth they'll work too. The manual doesn't recommend using the lower-end DX synths with DX Droid (that's DX9, DX21, DX100 models) but I'll bet they'll work to some extent, if you don't mind some unpredictability due to the different algorithms & operators.

The error-handling on DX Droid is impressive - in fact I couldn't find any kind of bugs anywhere in the program. I have a 1040ST with TOS on ROM, and from what I've heard the TOS on RAM-guys shouldn't have any problems either. There is no particular memory advantage to using a 1040ST with the DX Droid, since the total patches aren't in RAM. It runs on medium resolution on the color monitor (recommended) and on the high-res monitor as well. The only complaint I have is the hard-copy print-outs aren't so great - mostly just screen-dumps which my IBM Proprinter handles somewhat erratically. Still, the only use you would have for printouts is for somebody else to look at.

The "Preliminary Users Manual" that came with my DX Droid looked like it was fresh off the local high school's print shop press, but it was well-written and informative. Actually, you can pretty much file the manual away after you breeze through it - the program is so easy to use that you'll intuitively pick it up and become a genius synth programmer in no time at all.

All in all, this is the best MIDI software I've ever used, and I've used practically everything there is on the market. It's a must for any serious recording studio or musician. In fact, "DX Droid" should put to rest the idea that the Amiga is the ultimate art/music computer. Some may think it's a bit pricey at the $245 list price (for Atari software!?), but imagine how much it would cost on the IBM PC - if you could get it Those of you who do plunk down the bucks for "DX Droid" will never regret it. Best of all, it's FUN!! This alone is a new first for MIDI software. Now I can't wait to see what Hybrid Arts comes out with next!

MIKE SEELEY is a professional keyboard player from the Jersey Shore, whose credits include working with BonJovi, Surgin', Billy Squier & Dan Hartman, as well as the upcoming Billy Branigan LP on Polygram. He owns an 8-track recording studio as well as a MIDI software & synth programming company, and hopes to be helping more people get introduced to MIDI, computers and music.

His company, Livewire Audio, sells MIDI software (including Hybrid Arts) and synthesizer sound programs (or patches or files or....) and if you're interested in any MIDI software or hardware for any computer, or sounds for your synth, or if you're just curious about MIDI and computers, feel free to give him a call at (201) 870-3115.

Users groups wishing to reproduce this review must request permission from the author. He can be reached at the above phone number or you may write to him. His address is: Mike Seeley President Livewire Audio
79 Shrewsbury Ave. Oceanport, NJ 07757

Reprinted from THE ATARI JOURNAL April 1986

Public Domain Program Documentation (program included with disk subscription)

DOCUMENTATION FOR "FDI"
A FLOPPY DISK INDEXING PROGRAM
by Rod Waehner CIS [75156,3131]

OVERVIEW

FDI is a program to help you catalog your library of "floppy" disks and the files that reside on each disk. It does not use windows, drop-down menus, fancy colors, or any of that neat stuff. It does give you a complete index of the files you have, what disks they are on, when they were last updated, and lots of room for descriptions and comments. And, it is free!

SETTING UP

To keep things simple, you should keep your Floppy Disk Index on a disk by itself (or in a folder by itself on a hard disk). A single sided disk will have room for about 1,000 entries. To get started, copy FDI.TOS to an empty disk and double-click on it.

START UP

FDI will not run until it sees a valid system date. If the current system date is not valid, you will be asked to enter the current date, time, and day of week. Use three letter abbreviation for day of week (Mon, Tue, etc.) When all fields are correct press F10 to continue.

When prompted for "Filesystem name" enter the name you wish to assign to your catalog file. This name must start with a letter and be no more than eight characters long. Do not use an extension ("."). If the filesystem you select does not exist on the disk, an empty one will be created. A filesystem consists of three files: the data file with an extension of ".DT" and two index files with extensions of ".NX0" and ".NX1". These three files must be kept together when copied or backed up.

UP AND RUNNING

If you are starting out with an empty filesystem, the first thing to do is Index your first disk (option "I" for Index). CAREFULLY follow the prompts as to when to insert and remove disks from drive A. You should see a new screen:

DISK ID

FDI keeps track of your disks by a four character disk ID. You may use any method you like for assigning these ID's; as a suggestion, I use this: The first character is a letter indicating the contents or type of programs on the disk (B for BASIC, D for demos, L for LOGO, etc). The next three characters form a number to identify the disk.

Therefore, my BASIC disks are labelled B001, B002, etc. My LOGO disks are labelled L001, L002, etc.

The important thing to remember is that no two disks should be given the same ID. If you do not follow this rule, you are guaranteed to lose data. By default, FDI writes the ID you selected to the disk being indexed. If you do not wish to write that Disk ID on that disk, enter "n" in the field "Write ID to disk". If you allow FDI to write the ID to the disk it will write a file called "DISKID.QQQ" onto the disk. Whenever the disk is re-cataloged later, you will not be promted for the disk ID again.

When the Disk ID screen is complete, press F10 to continue.

MAIN SCREEN

Most of the options on the main screen are fairly self-explanatory (or safe to experiment with). However, some may need explanation:

Add -- This option will allow you to manually add a filename/disk to your catalogue. This may be used to index non-standard or copy-protected disks. Fill in the information on the screen. The F10 Function key will complete the 'add' option and return you to the bottom menu.

Beginning of file -- Choosing this option will take you to the beginning of the file. The 'beginning' of file will be dependent upon the prime index you are in (where the asterisk is located).

Delete this record -- The record that is currently displayed will be deleted. You will be prompted at the bottom of the screen - "Are you sure you want to delete this record? " Input "Y" if you are sure. Input "N" if you do not want to delete this record.

End of File -- This selection will take you to the end of your file (according to your current index) and will display that record.

Find -- This will allow you to "Find" a Disk name or Filename. You do not need to enter the full name you are searching for. For example, if your current index is "file name" and you enter "m", you will see the first file that begins with "m". If you enter "me", you will see the first file that begins with "me".

Index -- With this option you should follow the instructions in the "Up and Running" and "Disk ID" sections. You will use this selection whenever you need to read a new disk or read/index a modified disk to catalog in this program. Carefully follow the prompts when swapping disks.

List -- When you choose this option you will get another screen with Disk ID, File name, File Description, Date last modified, Time, and Size of the file. The options, which are presented on the bottom of the screen are : Next Page, Previous Page, UP ARROW, DOWN ARROW, and F10 to select.
If you wish to select a particular file/record from the List screen, use any of the above selections to get the cursor there, and press F10. That record will then display on the screen.

Next Entry -- This option will display the next record in the file, according to the current index.

Output to printer -- Printing will start at the record currently selected and continue until end of file or until you press the 'Q' key. The report is sorted by file name or by disk ID, depending on the current index.

Previous Entry -- This selection will allow you to view the previous entry of your records. As in all Next or Previous fields, it is alpha-based and dependent upon the Prime Index you have chosen.

Quit this program -- This will take you back to the GEM Desktop.

Switch prime index -- As you travel foreward or backward thru your file you see entries in sequence by either file name (the default) or by disk ID. The sequence ("current index") is shown by the "*" next to the field name. You can change the current index by pressing "s".

Update this entry -- With this selection, you will be able to modify or add to any of the fields that are present in the displayed record. A frequent use of this selection would be after indexing a disk - you might want to type in descriptions of the files.

View Disk Directory -- With this selection, you will see 'Closing' at the bottom of the screen. Following this, you will be prompted to insert the disk you wish View. It will show you the Filename, Date last modified, Time, and Size of each file on that disk.

WARNINGS AND SUGGESTIONS

Rule number one is: ALWAYS finish out of this program by using the 'Q' option. DO NOT power off or re-boot while updating; doing so will certainly confuse the database, scramble data, or wreak general havoc. Rule number two is: ALWAYS keep backups of your filesystem (name.DT, name.NX0, name.NX1). Remember, even if you are updating only one entry, your whole database is subject to loss! As your database expands (and you have more time and effort invested in it) you will want to keep more than one copy. If your system has enough memory (one meg or more) you may want to load your filesystem to a RAM disk for updates. Some of the functions require a lot of disk accesses, and things go much more quickly on a RAM disk (or hard disk).

KEYS ENABLED IN FDI:

Backspace -- This key will move the cursor backwards and at the same time delete the character you Backspace over.

Left Arrow -- This key will move the cursor back (left) from where it is currently positioned. It will not delete or overwrite the characters currently there.

Shift Left Arrow -- This key moves the cursor to the first position in the current field.

Right Arrow -- This key will move the cursor forward (to the right) from where it is currently positioned, without deleting the characters it is going over.

Shift Right Arrow -- This key moves the cursor to the last position in the current field.

Insert -- When you depress the Insert Key it will allow you to "Insert" a character in the positon of the cursor.

Space Bar -- This key will overtype the characters that are 'spaced over' with the space bar.

Clr Home -- This key will clear the contents of the entire field that you are currently in.

Return -- The Return key will move the cursor to the next input field.

Up Arrow -- This key moves the cursor to the previous field.

F10 -- The F10 function key generally finishes out of the current screen and/or enters the data into the file.

Delete -- The Delete key will allow you to delete the character(s) that are positioned at the cursor.

UNDO -- The UNDO key allows you to leave an update screen without changing the data in your file.

Public Domain Program Documentation (program included with disk subscription)

Tiny Stuff and Tiny View
by David Mumper

ST Picture Compression

Any ST user who has pictures knows that, no matter how good (or bad) they look, they always take roughly 32128 bytes of disk space. This can be quite discouraging when you realize that this means that you can only fit 10 pictures on a single sided disk. From NeoChrome 0.5, the memory is used as follows...

```
    4 bytes telling the resolution (0=low, 1=medium, 2=high)
   32 bytes for the colors
   12 bytes set aside for a file name (always "          .    ")
    2 bytes telling which colors are set for rotating (first is always $80)
    2 bytes telling the speed and direction of rotation
    2 bytes telling how long to rotate
   74 reserved bytes (for expansion)
32000 bytes of screen memory
-----
32128 total
```

A more efficient way of saving the pictures would be to do a compression on the screen memory so that, if there were repetative sections, it wouldn't be as long. An example of how to do this could be to look at the Micro-Illustrator (Koala pad) compaction for the 8-bit computers. A straight memory save of a Graphics 8 (or 15) screen would take up 7680 bytes. Saving a picture with the Micro-Illustrator format can result in a picture as small as 32 bytes (if the screen was blank). This type of compression could be applied to the ST as well resulting in very large memory savings. TINYSTUFF is an implementation of just such a compression algorhythm that produces an average compression of about 64%. Comparing it to the Micro-Illustrator format, it's smallest save would be 42 bytes.

USING THE COMPACTOR

To use the compactor program, place it on a freshly formatted disk and add some pictures. For best results, place half as many uncompacted pictures on the disk as you could actually fit (5 on single sided and 10 on double sided). Next, double click on the compacter program. It will then load and begin executing. The program will begin by asking you the formats for the pictures to convert. As an additional feature, you can convert Degas format pictures to Neo (or vice versa) as well as compacting or uncompacting them. After you have chosen the source and destination formats for the pictures, the program will begin loading all the pictures that match the source format from the same disk and pathname as the program. The program will load and display each picture that has the correct extender for that file type (i.e. '.NEO' for NeoChrome, '.TNY' for Tiny or '.PI?' for Degas). As the program displays the pics, it will convert them and save them back to the same disk / pathname as the program. Each of the pictures will have the same name as they start with but the extender will be changed to that of the destination format.

Having your picture collection in a compacted format is handy to save space but is a nuisance if you can't look at them that way. Therefore, TINYVIEW is a displayer program for pictures in Tipy format. To look at the pictures, just place as many of them on a disk with the displayer program as will fit then double click the display program. It will search the directory for the first filename with a '.TNY' extender and then start displaying them from there. It will load and display all the pictures on the disk, cycling back to the beginning when the end of directory is reached. Pressing the space bar will cause the program to pause and wait for another key to be pressed before going on to the next picture. To exit the program, press any key other than the space bar while not paused. The picture that is being loaded will be displayed and then the program will return to the desktop. Any pictures that have color rotation information, will have that saved with the picture when it's compacted and displayed by this program.

The file format of the compacted files is as follows...

```
  1 byte that tells which resolution and if there is rotation information
    (if there is rotation information it is stored as follows
    1 byte for the start and end color to rotate (start is high 4 bits
      end is low 4 bits)
    1 byte for direction and speed of rotation (negative for left
      rotation positive for right, actual number is the number of
      vertical blank interrupts to wait between each rotation)
  2 bytes for the number of revolutions to make)
  2 bytes that tell how many control bytes there are
  2 bytes that tell how many data bytes there are
 32 bytes that tell the computer the colors
3-10667 control bytes
2-32000 data bytes
  -----
42-32044 minimum-maximum size
```

The screen is scanned column by column, plane by plane (assuming low res format) as the program does the compaction.

If there is rotation information, the resolution has 3 added to it to signify this.

Control bytes are interpreted as follows...

```
X < 0 -- 1-127 unique words are taken from the data section
X = 0 -- two more words are taken from the control section that is a value
         128-32767 which is the number of times to repeat the next data word
X = 1 -- two more bytes are taken from the control section that is a value
         128-32767 of unique words to take from data section
X > 1 -- 1 byte of repetative information is taken from data section and
         repeated 2-127 times.
```

Due to this compression algorythm, I've been able to place as many as 95 pictures on a double sided disk. (That's an 850% improvement!! and not common) Average disks will hold roughly 20-30 pictures. So far I've seen many very nicely done pictures on bulletin boards and visiting friends, so hopefully, by allowing the average user the ability to keep more pictures on less disks, we'll all be able to see many more artists showing their abilities on the ST.

SUBSCRIPTION INFORMATION

This newsletter is published by PORTER SOFTWARE, the editor is Ralph Porter. Material presented is done so without warranty of any kind. The articles are not necessarily the opinions of Porter software or its staff. All information presented is intended for non commercial use of its readers and may be copied and used without restriction in the non commercial market. However permission for commercial use must be obtained from the publisher. Other Atari users groups may obtain exchange rights simply by making a reciprocal agreement with us. Credit for articles and information obtained from other users groups will be printed with the article.

ATARI and ST are trademarks of Atari Corp, CPM and GEM are trademarks of Digital Research Corporation, MSDOS is a trademark of Microsoft Corporation, PCDOS a trademark of I.B.M. and UNIX a trade mark of Bell Laboratories. All references to these products are trademarked and should be noted.

All newsletters and publications need funds to operate. They also need input from users. At the present time this newsletter is published 6 Times a year. The cost is $3.00 per issue which includes postage in the U.S.A. It is also available with a floppy disk for $7.50. Foreign orders must include additional postage.

If you can't afford this price or you would like free copies of the newsletter there is a way you can get them. Simply submit an article, a program review, or a project related to the Atari ST Computers. We will credit you with one issue for each article received. If the submission is on disk then will also include the disk version at no cost.

ATARI ST USERS NEWS CATALOG

PUBLIC DOMAIN DISK LIBRARY

Several New Disks have been added. Some of these duplicate programs in other library volumes and/or those on ASTUN disks. This is because some of them are disks supplied as disk of the month at various users group meetings. ASTUN felt that they should be made available as is without additional editing of material. Other disks contain completely new material. An astrrisk "*" represents a new listing.

BASICLOGO1 Includes final release of ST BASIC and several programing examples. Also has logo code samples.

C1 Contains several samples of C source code. Includes xmodem, and kermit file transfer protocals, several graphics demos and a program to convert degas pictures to neo. Also a printer driver program and a utility to format 40 track 5 1/4 inch disks all with C source code.

MISC1 Miscillaneous programs including the bouncing ball demo, utilities for a one meg system and one for transferring 8 bit atari files to the ST, plus an IBM disk format program and a utility to display the CPU register contents after a system crashes.

MISC2 Contains 30 different fonts (Not Degas). Three pictures, Three games, A utility to redefine the keyboard to DVORAK, a piano simulation and a shareware 1985 Income Tax Template for VIP Professional.

PIC1 Seventeen squeezed NeoChrome pictures.

PIC2 Twenty one squeezed NeoChrome pictures.

PIC Eleven squeezed NeoChrome and animatation demo pictures

PIC4 Fifteen squeezed Degas Pictures

PRG1 Includes two terminal programs and several graphics demos and an on screen calculator.

APPL1 Includes NeoChrome and ST WRITER with documentation. A program to customize ST writer for your printer, the Megaroid Game, a two disk copy program and a 97K ramdisk with assembly source.

STSIG Volume 1 - Disk Based Newsletter put out by ST*SIG
Includes Documentation on adding a real time battery backup clock, and using 5 1/2 inch disks on the ST. It includes demos on VIP and Sundog. Also included is stterm, a modem program, a users comments on the ST and important Atari Phone numbers

STSIG Volume 2 - Disk Based Newsletter put out by ST*SIG
Includes Neochrome and ten squeezed pictures, a 97K ramdisk, a list of ST Basic commands, Graphics demos and effect programs, squeeze and unsqueeze utilities and xmodem, a terminal program.

STSIG Volume 3 - Disk Based Newsletter put out by ST*SIG
Includes a 10K print spooler, Xmodem definitions, Utilities for a One Meg ST, Antic GEM tutorials, ST Basic, reviews on hippospell and quest plus more graphics demos and some basic demos.

* STSIG Volume 4 - Disk Based Newsletter put out by ST*SIG (Two Disks $9.00)
Includes Four Fonts, Comdex Information, Reviews of Degas, Final Word and Fourth4, Neochrome, a sector editor, squeeze and unsqueeze, megaroid, a basic sound demo and manufacture demos of cracked and modula2.

MIDIMONO1 A midi demo with songs, a graphics and picture demo for monochrome systems.

PIC5 Contains a christmas demo of pictures and songs by Audio Light and Antic. Works with or without a midi instrument.

* ACEPD 001 Contains 30 Non Degas fonts. A 1985 Tax Template, The 97K ramdisk and 10K print spooler. A piano simulation, the ST writer printer configuration program and two graphics demos.

* ACEPD 002 Contains stsqueeze and stunsqueeze, both stterm and sttalk modem programs, stbasic with an adventure game, mastermind game, empire game and a financial utility.

* ACEPD 003 Contains several ram disks (some work - some dont), two calculator accessories, Eliza (an artificial intelligence program), a text encription program, a modifed control pannel that fixes the clock reset on boot. Also a maze generator, an accessory to save any desktop image as a degas picture, ripcord (a game) a disk toolkit accessory, a program to print the disk directory on a printer and one to rename disk volume labels. Also included are some interesting graphics demos.

* ACE MAY 86 Includes basic games of cribbage and yahtzee, picswitch, a utility to convert pictures between Degas, Neochrome, the Macintosh and the Amega plus sample pictures. Also includes the new verson of squeeze and unsqueeze for tos in rom, a customizable ram disk, the tiny tool accessories, the xmodem program a floppy disk seek test, and some disk utilities.

* STUN JUNE 86 Includes a calculator, a freeram indicator and a puzzle as accessories. Also includes doodle, 4 basic programs including baccrat, empirel, journe and stmast. There is a two disk copy program, a label program, megaroids game, a midi demo, ripcord, printdir and volume utilities.

* STUN JULY 86 Includes Tini, a picture compression utility and slide show with 5 pictures. Also has Fujiboink, an interesting graphics demo, a basic spelling check program, and FDI, a floppy disk indexer and a desk accessory that sends disk directories to the printer.

* VIP 1,2 & 3 VIP Worksheet Templates (3 Disk set $12.00)
* ZOOMRACKS DEMO DISK (c) by QuickView Systems
* VIP DEMO DISK (c) VIP Technologies
* ATARI ST DEMO DISK (c) Atari Corperation

(Unless noted otherwise Public Domain Disks Listed above are available for A $5.00 copy and handling fee. The price includes the disk.)

ASTUN VOLUMES Priced at $3.00 for Hardcopy or $7.50 for Disk and Hardcopy. (ASTUN hardcopy subscribers only may order the disk for $5.00).

ASTUN JANUARY-FEBUARY 1986
Articles include Comdex 85, A C introduction, Documentation on NEOchrome A One megabyte system upgrade, A breif summary of all ST Basic Keywords, A list of ST products and vendors and more. Additional information only with the disk version include more detail on the one meg update, Disk utilities for a one meg system, and two ram disks. One for all users 97K bytes long and one 360K bytes for one meg systems. Also the bouncing ball demo the STTALK freeware version .97 communications package.

ASTUN MARCH-APRIL 1986

Articles include more ST products and vendors, A detailed Basic quick reference, Basic Programs to display ten colors at a time in medium resolution and a ploting routine. A basic ST memory map. A summary of the C programming language. Some tips on using the ST desktop, An ST Writer quick reference and short program reviews on Calendar, M-Disk, Flipside, Mudpies, Lands of Havoc and Mi-Term. Additional files on the disk version include Thirty ST fonts, A utility to compress and reexpand files. STTERM, a communications package with source code, A Lotus/VIP template for 1985 Income Taxes, A mouse driven calculator, a checkbook program in basic, a piano emulation, a graphics demo, a utility to customize ST Writer for non Epson printers and an itemized listing of files on the new public domain disks.

ASTUN MAY-JUNE 1986

Articles in this volume include program reviews for Sundog, Modula II and Zoomracks. A Basic Technical description of the internal workings of the ST computer. An overview of structured programing techniques. An open letter to Atari and it's distributors, and Information on ST Bios and Extended Bios Calls including descriptions of each, C program calls and Sample Assembly Language calls. Disk Version additions include two games (cribbage and yahtzee) in basic. A utility to convert pictures from the Amega, The Macintosh, Neochrome and Degas to either Neochrome or Degas. It also allows conversion of Degas pictures from one resolution to another. Two pictures from the Macintosh and one from the Amega is included for demo purposes. There is a program that will print a disk directory on your line printer. Several Desk Accessories are included. A calculator, a calendar, a game puzzle, A disk, sector and memory editor, a utility to set colors for your desktop without the memory overhead required using the standard accessories and a utility that will save any CRT image to a deagus picture file. Also included is a formated printer driver program, a ram disk that may be customized to your desired size, A disk formating utility that will format disks using forty or eighty tracks, single or double sided, and nine or ten sectors per track. A utility to examine and change the stepper rates of your disk drive. These utilities are great if you hook up an IBM compatible disk drive. Finally new versions of the ST squeeze and unsqueeze utilities that have been fixed for TOS in ROM.

ASTUN JULY-AUGUST 1986

Includes several tips on ST operation, Program reviews on music studio, and DX Droid, plus product descriptions of DFT, KISSED, and DOS Shell. Articles explain the operation of the midi port and hardware documentation of the hard disk port. An announcement of the creation of STUN a network of clubs, newsletters and bbs's dedicated to support of the Atari ST computer line. Anouncements of new atari products and programs. A utility for hayes compatible modems to disable auto answer and some sample C code showing how to get around some I/O problems in the development system. A utility program also demonstrates how to make entering dates into a program more user frendly. Documentation for two programs on disk is also printed. One is FDI, A floppy disk indexing program, The other TINY, a utility to reduce the size of picture files and to disply them without having to expand the files. The disk includes these programs plus chgfor, a program which converts first word files into standard ascii files, and fujiboink, an interesting graphics demo taken from the Atari bouncing ball and a desk accessory that prints disk directories on a line printer.

To Order any of these items send check or money order to:

ASTUN (Atari ST Users News)
6157 South 700 West
Murray,Utah 84123

ASTUN PRICE LIST

ASTUN (Atari St Users News) is published 6 times yearly.
Hardcopy Subscriptions are $15.00 per year. Single Issue is $3.00
Disk plus Hardcopy Subscriptions are $40.00 per year. Single Issue $7.50.

The newsletter contains up to date information on the Atari ST computer line. It contains programming tutorials, product announcements and reviews. It is a public domain newsletter with most of the information submitted by Atari ST users.

The newsletter also sponsors a public domain library.

Disks are available for a copy and handling charge of $5.00 per volume including media and postage.

Picture disks are compressed. A utility is supplied to unsqueeze them on each disk. If you desire your disks uncompressed, specify when ordering. This will require two disks for each picture library disk. The price will be the same as two separate disks.

To order use the disk label. ie. misc1, bas-log1 or pic4

ASTUN ORDER FORM

Name _____

Address _____

City, State, Zip _____

Quantity	Product Number and/or description	Unit Price	Totol Price
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	____ASTUN FULL SUBSCRIPTION_____	_$40.00/yr_	_____
_____	____ASTUN DISK SUBSCRIPTION_____	_$30.00/yr_	_____
_____	____ASTUN HARDCOPY SUBSCRIPTION _	_$15.00/yr_	_____
_____	____ASTUN UPGRADE SUBSCRIPTION___	_Price Dif_	_____
Prices Include Shipping and Handling.		TOTAL	_____

Mail your order to

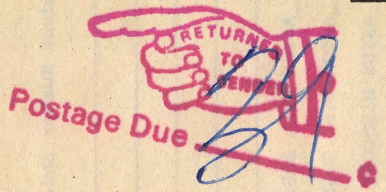
ASTUN (Atari ST Users News)
6157 South 700 West
Murray, Utah 84123

ATTENTION USERS GROUP OFFICIALS

This is a complimentary copy of ASTUN. Future copies of our newsletter will be sent free to all groups that are willing to exchange their newsletters with us.

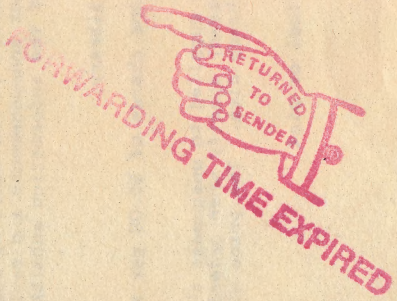
A S T U N
ATARI ST USERS NEWS
6157 South 700 West
Murray, Utah 84123

Bulk Rate
U.S. Postage
PAID
Midvale Utah
Permit No. 628



3

Sentient Atari Group Of Eureka (SAG
5226 Meyers Ave.
Eureka, CA 95501



Tried again
expired
12/17

Forward & Return Guaranteed